

## TRABAJO FINAL DE GRADO

**Grado en Ingeniería Electrónica Industrial y Automática**

# **BATTERY MANAGEMENT SYSTEM (BMS) PARA UN COCHE DE FORMULA STUDENT**



## **Volumen I**

## **Memoria**

**Autor:** Lucas Bouzón Pousa  
**Director:** Ángel Cuadras  
**Convocatoria:** Junio 2017



## Resum

Aquest treball consisteix en la realització de un BMS que es capaç de mesurar tots els voltatges i temperatures de les cel·les que componen una bateria, la intensitat d'aquesta, balancejar les seves cel·les i estimar el seu estat de càrrega (SoC) a partir d'una combinació de l'*open circuit voltage* (OCV) i del Coulomb Counting. S'ha realitzat sobre una bateria de 12 cel·les de  $\text{LiFePO}_4$ , però s'han establert les bases per ampliar el BMS i que sigui compatible amb una bateria d'un cotxe de Formula Student.

S'han fet diferents test de les mesures, de balanceig, de descarrega i de estimació SoC. Aquest últims dos assajos han sigut possibles gracies al disseny i construcció d'un sistema per descarregar la bateria a diferents intensitats per comprovar el correcte funcionament del BMS.

El sistema final permet mesurar correctament les tensions i temperatures i fa el balanceig passiu. Pel cas dels corrents els mesura amb un cert error que influeix en la estimació del SoC a partir del Coulomb Counting encara que el OCV funciona correctament.

Finalment, s'han proposat vàries millores tant en la realització de les mesures com en l'estimació del SoC com en possibles funcionalitats que podria tenir un futur BMS.



## Resumen

Este trabajo consiste en la realización de un BMS que es capaz de medir todos los voltajes y temperaturas de las celdas que componen una batería, la intensidad de ésta, balancear sus celdas y estimar su estado de carga (SoC) a partir de una combinación del *open circuit voltage*(OCV) y el Coulomb Counting. Se ha realizado sobre una batería de 12 celdas de  $\text{LiFePO}_4$ , pero se han establecido las bases para ampliar el BMS y que sea compatible con una batería de un coche de Formula Student.

Se han hecho diferentes tests de las medidas realizadas, de balanceo, de descarga y de la estimación del SoC. Estos dos últimos tests han sido posibles gracias al diseño y fabricación de un sistema para descargar la batería a diferentes intensidades para comprobar el correcto funcionamiento del BMS.

El sistema final permite medir correctamente el voltaje y la temperatura y hace el balanceo pasivo. En lo que concierne a la medida de corriente, hay un error que se ve reflejado en la estimación del SoC a partir del Coulomb Counting aunque el OCV funciona correctamente.

Por último, se han propuesto varias mejoras tanto en la realización de medidas como en la estimación del SoC como en posibles funcionalidades que podría tener un futuro BMS.



## Abstract

This project is based in the design and fabrication of a BMS capable of measuring all the voltages and temperatures of the battery cells, the current of the battery, balancing all the cells and estimating the State of Charge (SoC) from a combination of the open circuit voltage method (OCV) and the Coulomb Counting. The BMS is designed to monitor a battery made of 12 LiFePO<sub>4</sub> cells but the basis to make the system expandable and suitable for a Formula Student car battery have been established.

Some tests have been made about the measurement, the balancing, the discharge of the battery and the SoC estimation. These last two tests have been possible thanks to the designed and manufactured system of the battery discharge to ensure the correct working of the BMS.

The final system measures correctly the voltages and the temperatures and makes a passive balancing. Although the OCV works properly, the current measurement has a little error that influences in the SoC estimation by the Coulomb Counting.

Finally, some improvements about the measurements have been proposed, the SoC estimation and the possible extra functionalities a BMS may have.





## Agradecimientos

Me gustaría agradecerles a mis padres y a mi hermano, Adriana Pousa, Pablo Bouzón y Gonzalo Bouzón todo el soporte moral que me han aportado durante la realización del TFG. Doy gracias a mis amigos por los ánimos y la motivación proporcionada.

También debo agradecer a todos los integrantes del equipo de Formula Student, e-Tech Racing por la ayuda técnica recibida. Agradecer, por otra parte, a e-Tech Racing como entidad por aportar material necesario para la realización de este trabajo.

Por último, me gustaría darle las gracias a mi tutor del TFG, Àngel Cuadras por la implicación que ha tenido en el trabajo.



# Índice

<b>RESUM</b>	<b>I</b>
<b>RESUMEN</b>	<b>III</b>
<b>ABSTRACT</b>	<b>V</b>
<b>AGRADECIMIENTOS</b>	<b>VII</b>
<b>1. PREFACIO</b>	<b>1</b>
1.1. Origen del trabajo .....	1
1.2. Motivación .....	1
1.3. Requerimientos previos.....	1
<b>2. INTRODUCCIÓN</b>	<b>3</b>
2.1. Objetivos del trabajo.....	4
2.1.1. Objetivo 1: Diseño del BMS.....	4
2.1.2. Objetivo 2: Estimación del State of Charge (SoC) .....	5
2.1.3. Objetivo 3: Comunicaciones.....	5
2.1.4. Objetivo 4: Diseño de sistema para simular la descarga de la batería .....	5
2.1.5. Objetivo 6: Futuros estudios .....	5
2.2. Alcance del trabajo .....	5
<b>3. STATE OF THE ART</b>	<b>7</b>
3.1. Automóvil eléctrico .....	7
3.2. Baterías.....	8
3.2.1. Baterías de Li-ion .....	10
3.3. BMS .....	11
3.3.1. Balanceo de celdas .....	12
3.3.2. Calculo de parámetros característicos.....	15
3.3.3. Topología .....	17
<b>4. DISEÑO DEL BMS</b>	<b>21</b>
4.1. Diseño del hardware .....	21
4.1.1. Batería.....	21
4.1.2. Obtención de voltajes y balanceo .....	23

4.1.3.	Medición de temperatura.....	26
4.1.4.	Medición de corriente .....	31
4.1.5.	Can hardware .....	32
4.1.6.	Microcontrolador .....	32
4.1.7.	Funcionalidades extra .....	33
4.1.8.	Diseño del sistema de descarga.....	34
4.2.	Diseño del software .....	36
4.2.1.	Programa mínimo de obtención de medidas .....	37
4.2.2.	Estimación del SOC .....	42
<b>5.</b>	<b>FABRICACIÓN .....</b>	<b>51</b>
<b>6.</b>	<b>RESULTADOS .....</b>	<b>57</b>
6.1.	Resultados de la prueba de medición de voltajes .....	57
6.2.	Resultados de las mediciones de temperatura.....	58
6.3.	Pruebas mediciones de corriente.....	59
6.4.	Pruebas finales de las mediciones.....	61
6.5.	Pruebas de balanceo .....	62
6.6.	Pruebas de la estimación del SoC.....	64
6.6.1.	Estimación del SoC a partir del OCV .....	64
6.6.2.	Estimación del SoC a partir del Coulomb Counting .....	66
6.7.	Prueba de descarga .....	66
6.8.	Comparación de la impedancia de una celda a diferentes temperaturas .....	68
6.9.	Envejecimiento de la celda con la realización de los test del OCV .....	69
<b>7.</b>	<b>ANÁLISIS DEL IMPACTO AMBIENTAL .....</b>	<b>71</b>
<b>8.</b>	<b>FUTUROS ESTUDIOS .....</b>	<b>73</b>
8.1.	Mejoras en las medidas.....	73
8.2.	Mejoras de la estimación del SoC .....	74
8.3.	Ampliar el sistema para un mayor número de celdas .....	74
8.4.	Posibles mejoras adicionales.....	75
	<b>CONCLUSIONES .....</b>	<b>77</b>
	<b>PRESUPUESTO Y ANÁLISIS ECONÓMICO .....</b>	<b>79</b>
	<b>BIBLIOGRAFÍA .....</b>	<b>85</b>
	<b>ANEXO A .....</b>	<b>89</b>

A1.	Esquemático general de la PCB diseñada.....	89
A2.	Esquemático detallado de la placa DC2260A.....	92
A3.	<i>Layout</i> de la PCB.....	96
A4.	Programa del microcontrolador .....	98



# 1. Prefacio

Este trabajo consiste en el diseño, fabricación y testeo de un Battery Management System, desde ahora en adelante BMS, para un coche de Formula Student. La función de este sistema es monitorizar y controlar que las celdas de una batería trabajen dentro del llamado “*Safe Operation Area*”, es decir, dentro de unos límites de voltaje, intensidad y temperatura. Si alguno de estos parámetros no está dentro de la “*Safe Operation Area*”, las celdas de la batería pueden perder vida útil, capacidad o llegar, incluso, a ser peligrosas si se producen reacciones químicas indeseadas.

## 1.1. Origen del trabajo

La idea de este trabajo nace de las necesidades y deseos de e-Tech Racing, equipo de Formula Student de la facultad de Escola d’Engineers de Barcelona Est (EEBE) de la Universitat Politècnica de Catalunya (UPC). Dicho equipo diseña y construye cada año un monoplaça eléctrico para participar en la competición de Formula Student. Prácticamente toda la electrónica del vehículo está diseñada por integrantes del equipo, exceptuando varios sistemas, entre los cuales el BMS. Actualmente, el BMS que se ha estado usando los dos últimos años para el coche de e-Tech Racing es el modelo Freesafe-XT de la empresa Freemans. Este producto ha provocado ciertos contratiempos al equipo, ya que, al ser un sistema comercial, se dependía de la empresa (que no siempre ha cumplido los plazos de entrega) para cualquier modificación del firmware. Esto hace surgir la idea de diseñar un BMS específico para el equipo y, aparte de solucionar la dependencia de una empresa externa, daría puntos extra en la competición.

## 1.2. Motivación

La motivación principal del trabajo es aprender todo lo posible sobre la gestión de baterías de vehículos eléctricos y demostrar los conocimientos adquiridos tanto en la carrera como en e-Tech Racing. Además, un valor añadido muy importante es poder aportar al equipo un BMS que puedan entender, reprogramar, rediseñar, mejorar y modificar y que, además de funcionar perfectamente, sirva para ganar puntos en las competiciones.

## 1.3. Requerimientos previos

Partiendo de la base, que los conocimientos en electrónica ya se han obtenido debido a que es un trabajo final de grado, uno de los requerimientos previos más importantes era entender que era un

BMS. Se debía entender su complejidad, su viabilidad de hacerlo para un estudiante de grado y qué conocimientos eran necesarios para hacerlo.

Por otra parte, el equipo tenía sus propios requerimientos, en cuanto al BMS que se diseña. Este tenía que cumplir unas funciones específicas y cumplir con la normativa de la competición. Estas funciones son las siguientes:

- Ser específico para baterías hechas con celdas de Li-ion.
- Monitorizar el voltaje y temperatura de todas las celdas.
- Permitir tanto carga como descarga y medir la corriente.
- Ser capaz de hacer algún tipo de balanceo.
- Ser capaz de estimar el State of Charge.
- Comunicarse con el resto de electrónica del coche vía el bus CAN para enviar la información sobre el estado de la batería.
- Comunicarse con el ordenador.



## 2. Introducción

La contaminación ha provocado la necesidad de realizar un cambio energético, reduciendo el uso de las energías no renovables y apoyando, desarrollando e impulsando las energías renovables. Dicho cambio se ve reflejado en casi todos los sectores, entre ellos, la automoción; las compañías invierten e investigan cada vez más en la movilidad eléctrica y sostenible. Por este motivo también, la competición de Formula Student, antiguamente solo de coches de combustión, ahora ya tiene la categoría de vehículos eléctricos.

Ya que este tipo de vehículos se basan en un sistema de baterías eléctricas que almacenan la energía, se necesita tener una serie de conocimientos básicos sobre éstas para entender un poco mejor el objetivo de este trabajo. Las baterías están compuestas por asociaciones de celdas en serie y paralelo, dependiendo del voltaje y capacidad necesarios para cada aplicación. Estas celdas son los sistemas electroquímicos que trabajan a un voltaje característico de la reacción química de la celda y que proporcionan corriente eléctrica para alimentar el circuito externo.

Hay diferentes tipos de químicas de celdas. Las más comunes son las de plomo, Niquel-Metal-Hidruro (NiMH) y litio. En este proyecto se trabaja con celdas de litio (Lithium-Ion cells). Este tipo de celdas es el más extendido actualmente en aplicaciones móviles, y en particular en la automoción, debido a su alta densidad de energía, larga vida, y su permisividad de descarga a altas intensidades. El problema de éstas es que son muy sensibles al punto de operación, es decir, cuando no trabajan en el ya mencionado, “Safe Operation Area”, pierden cualidades, como, por ejemplo, capacidad o ciclos de vida. Es por ello que existe la necesidad de tener un sistema que las controle, el BMS.

Este sistema normalmente también tiene funcionalidades extras a parte de la de monitorizar la batería. La más común es la de permitir un balanceo de las celdas de la batería para evitar desequilibrios de voltaje entre ellas. Otra de esas funciones es la de comunicarse con el resto de sistemas electrónicos del coche y con un ordenador externo. Esto permite que estos sistemas puedan actuar en función del estado de la batería, y, también permite a una persona visualizar, en tiempo real, dicho estado a través de un ordenador. Por último, el BMS suele ser capaz de estimar ciertos parámetros de la batería como el State of Charge, entre otros. Este parámetro es muy importante ya que te indica en porcentaje la cantidad de capacidad que le queda a la batería.

Dado que una de las bases de este trabajo es la Formula Student y el equipo e-Tech Racing, se va a explicar brevemente dicha competición y dicho equipo. Formula Student es un evento que se celebra en 16 países y que reúne a universidades de todo el mundo para competir entre ellas. El objetivo de la competición es el diseño y construcción de un monoplaça. Hay tres categorías: combustión, eléctrico y driverless. Es una competición ingenieril, en la que no se buscan los mejores pilotos sino los mejores

ingenieros. Por eso, las puntuaciones se dividen en dos eventos: los estáticos y los dinámicos. En los estáticos, los equipos defienden el diseño de su monoplaça, sus costes y hacen un business plan para vender el coche. Antes de pasar a los eventos dinámicos, el coche pasa el “scrutineering”, donde los jueces verifican que los coches cumplen con la normativa. Los eventos dinámicos tienen varias pruebas: la aceleración, en una recta de 75 metros; el skipad, un circuito en forma de 8 que sirve para comprobar las fuerzas laterales; el autocross, una vuelta rápida a un circuito de aproximadamente 1 km de longitud; la endurance, una carrera de 22 km en el mismo circuito del autocross; y, finalmente, la efficiency, donde se comprueba la energía consumida por los equipos durante la endurance. e-Tech Racing es el equipo de Formula Student de la EEBE (anteriormente EUETIB). Este equipo es relativamente nuevo, nació hace 4 años. Su primer coche no se llegó a construir debido a falta de presupuesto y conocimientos; el segundo llegó a la competición de España, pero no pasó “scrutineering”; y, el último, el tercero compitió en todas las pruebas, exceptuando la aceleración e hizo un buen papel en la competición.

## 2.1. Objetivos del trabajo

El objetivo final del trabajo es proporcionar los conocimientos suficientes para obtener un BMS totalmente funcional para el coche de e-Tech Racing. Debido a varios factores como el presupuesto, la no disponibilidad de celdas y la complejidad, en este trabajo no se hará el sistema completo para la batería del coche, sino que se hará un módulo basado en una batería formada por 12 celdas proporcionadas por el equipo. Dicho módulo podrá ser funcional para cualquier tipo de batería de Li-ion y cumplirá las reglas de la competición.

### 2.1.1. Objetivo 1: Diseño del BMS

Como se ha dicho anteriormente se diseñará y construirá un módulo que controlará una batería de 12 celdas. Este módulo deberá cumplir las siguientes funciones:

- Medir todos los voltajes y temperaturas de las celdas que controle.
- Ser capaz de balancear las celdas mediante un balanceo pasivo.
- Medir la corriente del conjunto.
- Permitir tanto descarga como carga.

### **2.1.2. Objetivo 2: Estimación del State of Charge (SoC)**

Actualmente no existe un cálculo sencillo exacto para estimar el State of Charge (SoC) mencionado anteriormente. Por tanto, se tendrá que hacer un estudio para ver cuál es la forma óptima de estimarlo y como corregir los posibles errores.

### **2.1.3. Objetivo 3: Comunicaciones**

Como se ha dicho anteriormente, una de las funciones más comunes de un BMS es la capacidad de comunicarse. Por tanto, el BMS diseñado en este trabajo deberá ser capaz de dichas comunicaciones, mediante el protocolo CAN para la transmisión de información entre sistemas electrónicos del coche y mediante el protocolo RS-232 para la comunicación con un ordenador externo.

### **2.1.4. Objetivo 4: Diseño de sistema para simular la descarga de la batería**

Se tendrá que pensar, diseñar y construir un sistema capaz de funcionar como carga para que la batería se descargue y verificar así el correcto funcionamiento del BMS.

### **2.1.5. Objetivo 6: Futuros estudios**

Gracias al conocimiento obtenido a la realización del trabajo se tendrán que marcar las bases para continuar mejorar este proyecto: hacerlo compatible con más módulos para controlar más celdas, mejorar la estimación del SoC, mejorar la obtención de datos, mejorar la comunicación, mejorar la visualización de datos o dar ideas de posibles funcionalidades extra.

## **2.2. Alcance del trabajo**

El alcance final de este trabajo es hacer un BMS para la batería de e-Tech Racing partiendo de la base y conocimientos que se presentan en este proyecto. El modulo diseñado en este trabajo podrá funcionar perfectamente en las baterías de e-Tech Racing, pero dado que la batería está compuesta por muchas más de 12 celdas se tendrá que adaptar el modulo y fabricar más para poder hacer un BMS completo.

Así como para la temporada 2016-2017 se decidió, desde un principio, en mantener el BMS comercial que se había estado usando la pasada temporada, para la temporada 2017-2018, gracias a este trabajo, se realizará un BMS propio. Esto permitirá al equipo conocer perfectamente el sistema, ser capaz de modificar o cambiar tanto el software como el hardware sin ningún problema y, además, dará muchos puntos en el Design Event de las competiciones en las que se participe.



### 3. State of the Art

Este capítulo dará una idea general del estado actual de la tecnología que comprende este trabajo. Primero se comentará el estado del vehículo eléctrico ya que el objetivo final es la implementación del BMS en un coche eléctrico de competición. Posteriormente se expondrán los tipos de baterías que usan dichos vehículos y se compararán sus características. Finalmente, entrando ya entrando directamente en el tema del trabajo, se especificarán las funcionalidades del BMS.

#### 3.1. Automóvil eléctrico

Aunque a lo largo de la historia ha habido varios modelos de automóvil eléctrico lo cuales nunca han tenido mucho éxito hasta el siglo XXI. En 2008, más o menos, el aumento de precio del petróleo y la necesidad de reducir la emisión de los gases de efecto invernadero provocados por la combustión de la energía fósil concienciaron al mundo de la necesidad del cambio energético mencionado anteriormente. Dicho cambio requiere de una movilidad más sostenible, reduciendo la contaminación. Unos de los motivos que han impulsado la movilidad eléctrica como solución han sido los avances en las tecnologías de la batería y la mejora de la gestión de energía [1].

Por todo ello, se ha investigado tanto en automóviles híbridos como eléctricos para mejorar su eficiencia y su autonomía, entre otras cosas. Los híbridos se propulsan combinando un motor de combustión con un motor eléctrico. De esta manera la energía necesaria para mover el coche no solo provenía de la energía fósil, sino que también de la energía eléctrica por lo que se reducía el consumo de gasolina, y consecuentemente la contaminación. De estos vehículos hay de dos tipos: el no enchufable y el enchufable. En el no enchufable, parte de la energía generada por el motor que se necesita para recargar la batería eléctrica del coche, se usa para alimentar un motor eléctrico. En este tipo de coches el motor eléctrico se usa básicamente como complemento del de combustión para reducir un poco el consumo y el gran inconveniente que tienen es que la autonomía en modo solo eléctrico es de muy pocos kilómetros, menos de 10. El coche híbrido enchufable, es un poco diferente, ya que su batería de alto voltaje es más grande y aparte de recargarse por la energía producida por el motor de combustión y de la frenada regenerativa, permite ser recargada por un cargador externo. De esta manera consigue más almacenamiento de energía eléctrica para el motor eléctrico y, por tanto, tiene una autonomía, en modo totalmente eléctrico, considerablemente superior al no enchufable llegando a recorrer en modo totalmente eléctrico hasta 50 km y, en el caso de que se usen los dos al mismo tiempo, reduce en una proporción mayor el consumo de combustible. Por último, el coche eléctrico se podría considerar como una evolución del híbrido enchufable, ya que elimina el motor de combustión e aumenta la capacidad de las baterías [2].

Hoy en día, se cree que los coches eléctricos irán sustituyendo tanto a los coches de combustión como a los híbridos. Los coches eléctricos tienen grandes ventajas respecto a los de combustión: no contaminan, ofrecen el par motor instantáneo, no necesitan cambios de marchas, pueden llevar más de un motor para mejorar el control del coche y, por último, normalmente las baterías que son lo más pesado se sitúan en la parte inferior del coche disminuyendo el centro de gravedad y, por tanto, aumentando la estabilidad y la seguridad. Como todo, también tiene inconvenientes: el primero es la autonomía, que es bastante inferior a la autonomía de la de un coche de combustión, aunque, hoy en día, hay modelos que llegan hasta los 600 km de autonomía; otro inconveniente es el tiempo de recarga, a diferencia de los coches de combustión que en dos minutos rellenan el depósito de gasolina, las baterías de un coche eléctrico necesitan varias horas para recargarse correctamente, pero actualmente ya existen baterías que permiten recargas rápidas parciales para, por ejemplo, tener una autonomía de 100 km con solo 20 minutos de carga. El último inconveniente es el precio, es objetivamente más caro hacer un coche eléctrico a uno de combustión debido, en gran parte a la fabricación y estudio de las baterías [3].

### 3.2. Baterías

Como se ha expresado anteriormente, uno de los puntos más claros a mejorar en el mundo del automóvil eléctrico es el de las baterías. Estas son, en términos generales, asociaciones de celdas, que son el elemento que almacena la energía eléctrica. Dichas asociaciones dependen de las necesidades de su uso: pueden ser en serie, aumentando así el voltaje total de la batería, en paralelo aumentando la capacidad total de la batería o una combinación de ambas para aumentar tanto el voltaje como la capacidad.

A lo largo de los años el tipo de celda usado para los automóviles eléctricos ha ido cambiando. Los primeros prototipos de coches eléctricos están datados a finales del siglo XIX y utilizaban una batería basada en celdas de plomo-ácido. A principios del siglo XX, dado que estas proporcionaban muy poca autonomía: entre 25 y 50 km, surgieron las baterías basadas en celdas de níquel-hierro (NiFe) que proporcionaban una autonomía de 130 km a 32 km/h. Posteriormente, durante el siglo XX los coches eléctricos dejaron de fabricarse en masa, exceptuando algún que otro modelo esporádico debido a los avances del motor de combustión, que permitía autonomías mucho mayores y recargas de energía mucho más rápidas. En 1999 uno de esos modelos esporádicos, el GM EV1 de segunda generación implementó un nuevo tipo de celdas de níquel e hidruro metálico (Ni-MH) que permitía una autonomía de entre 160 km y 225 km, almacenando una energía total de 26,4 kWh, con una densidad energética de 140 Wh/l y una energía específica de 60 Wh/kg. Este modelo se retiró a los pocos años de su lanzamiento y ya en el siglo XXI aparecen las baterías de iones de litio (Li-ion), con modelos de coches como el Nissan LEAF o el Tesla Roadster. Debido, entre otras cosas, a su densidad energética y energía

especifica superiores a 250 Wh/l y 100 Wh/kg, respectivamente, este tipo de celdas es la más utilizada actualmente para fabricar las baterías de los vehículos eléctricos. Se ha llegado a obtener una autonomía total de 660 km con una batería de 100 kWh y aceleraciones de 0 a 100 km/h por debajo de los 3 s con este tipo de celdas. Hoy en día, aunque se están investigando otro tipo de químicas, las celdas de iones de litio son las más utilizadas y las que formarán las baterías de los coches de por lo menos los 2 o 3 años siguientes [4]. La Tabla 1 muestra la comparación de las especificaciones más características de las baterías que se han ido nombrando.

Tabla 1 Comparación de los tipos de celda que se han usado a lo largo del tiempo para los coches eléctricos [5], [6]

Tipo de celda	Plomo-ácido	NiFe	Ni-MH	Li-ion
Energía específica	30-50 Wh/kg	30-50 Wh/kg	60-120 Wh/kg	90-250 Wh/kg
Ciclos de vida	200-300	1000	200-500	500-2000
Autodescarga	5 %/mes	20 %/mes	30 %/mes	5 %/mes
Voltaje máximo	2,4 V	1,6 V	1,6 V	4,2 V
Voltaje mínimo	1,75 V	1 V	1 V	2,5 – 3 V
Voltaje nominal	2 V	1,2 V	1,2 V	3,3 – 3,7 V
Máxima corriente de descarga	5 C	10 C	5 C	2-30 C
Temperatura de carga	-20 a 50 °C	0 a 45 °C	0 a 45 °C	0 a 45 °C
Temperatura de descarga	-20 a 50 °C	-20 a 65 °C	-20 a 65 °C	-20 a 60 °C
Requerimientos de seguridad	Térmicamente estable	Térmicamente estable, protección de corriente	Térmicamente estable, protección de corriente	Necesario circuito de monitorización y protección, el BMS

Cabe destacar que, aunque la tecnología de las baterías haya avanzado considerablemente, siguen sin ofrecer las características de la energía fósil. El diagrama de Ragone (Figura 1) muestra la relación entre la energía específica y la potencia específica de diferentes tipos de baterías. Además, se han marcado

los objetivos de estas dos características para los vehículos híbridos, para los híbridos enchufables y para los eléctricos. Se puede observar que para los dos tipos de coches híbridos los objetivos se pueden cumplir pero que para los coches eléctricos todavía no se cumplen ni con las baterías de Li-ion.

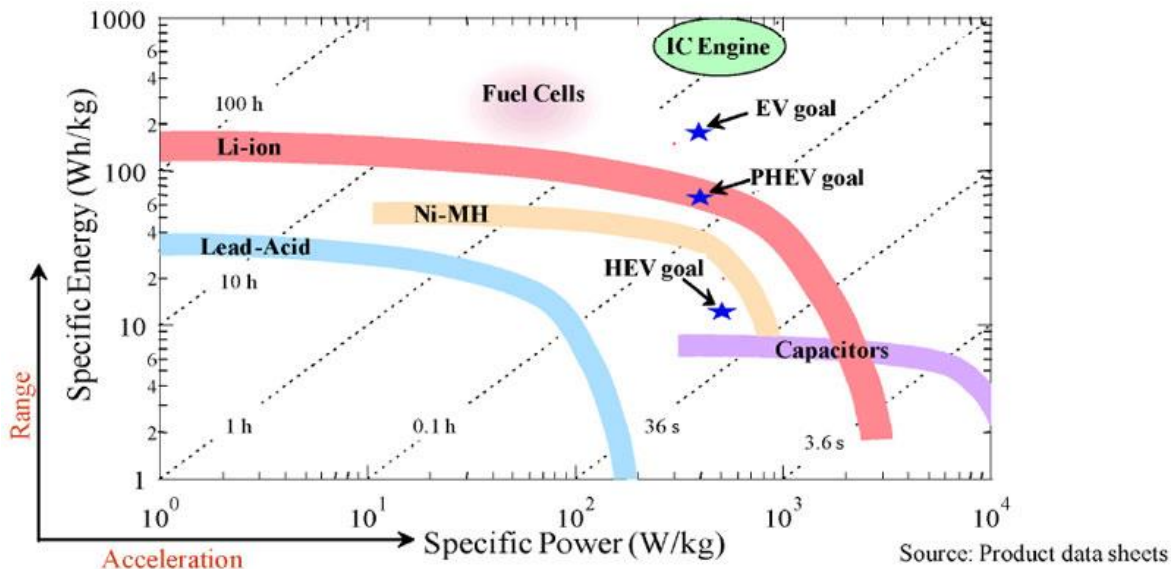


Figura 1 Diagrama de Ragone [7]

### 3.2.1. Baterías de Li-ion

Dentro de las celdas de iones de litio hay diferentes variantes dependiendo del material iónico que las forme. De ahí, que la Tabla 1 muestre amplias características de este tipo de celdas.

Las celdas compuestas por litio óxido de magnesio ( $\text{LiMn}_2\text{O}_4$ ) son una variante de las celdas de Li-ion que se utilizan para herramientas eléctricas o aparatos médicos. Una evolución de estas son las celdas de litio níquel magnesio óxido de cobalto ( $\text{LiNiMnCoO}_2$ ) o las celdas de litio níquel cobalto óxido de aluminio ( $\text{LiNiCoAlO}_2$ ). Estas dos variantes consiguen tener una gran energía específica, permitiendo también grandes intensidades de descarga. Por lo que las hacen ideales para ser utilizadas para hacer las baterías de los coches eléctricos. Además, también se utilizan para aplicaciones industriales.

También existen celdas de litio titanito ( $\text{Li}_4\text{Ti}_5\text{O}_{12}$ ) que proporcionan poca energía específica, pero que son una de las más seguras y la que proporciona el mayor número de ciclos de vida.

Las celdas de litio fosfato de hierro ( $\text{LiFePO}_4$ ) son muy estables y permiten grandes descargas. Aunque tengan poca energía específica comparada con otras celdas de Li-ion, sus dos características diferenciales las hacen útiles para aplicaciones en los vehículos eléctricos. Muchos equipos de Formula Student optan por este tipo de celdas para formar las baterías del monoplaza.



Por último, las celdas compuestas por litio óxido de cobalto ( $\text{LiCoO}_2$ ) son las que tienen la energía específica más alta, pero es la más peligrosa debido a su inestabilidad. Aunque estas se usen, en gran parte, en dispositivos portátiles, también se han llegado a utilizar en vehículos eléctricos de competición, como las que usa e-Tech Racing.

Cabe destacar que la elección de una variante u otra de las celdas de Li-ion depende en gran parte de su aplicación. Es decir, por ejemplo, si el objetivo es alimentar un motor que funciona intermitentemente con grandes consumos de corriente se necesitara una batería que permita dichas corrientes, como por ejemplo las celdas de  $\text{LiFePO}_4$ . Por el contrario, si lo que se busca es optimizar el espacio y el peso, se necesitara una gran densidad de energía como las que proporcionan las celdas de  $\text{LiCoO}_2$  [8].

Aunque las baterías formadas por celdas de Li-ion proporcionan unas características remarcables comparadas con otro tipo de celdas, estas necesitan un sistema que las monitorice y controle que trabajen correctamente. Dicho sistema es el, ya mencionado, BMS.

### 3.3. BMS

El Battery Management System es un sistema electrónico que se encarga de monitorizar el estado de las celdas de una batería. Es un sistema pensado para ser utilizado en las baterías formadas por celdas de Li-ion para controlar que estas trabajen en el “Safe Operation Area”. Dicha área está comprendida por unos límites de voltaje, corriente y temperatura. Si la celda no trabaja dentro de estos límites podría perder capacidad, ciclos de vida e incluso, podría ser peligrosa provocando reacciones químicas indeseadas. Por tanto, la función del BMS es medir continuamente estos parámetros y mediante un relé, un mensaje o una señal digital controlar el funcionamiento de la batería, es decir, permitir que esta se cargue o descargue si todos los parámetros están dentro del “Safe Operation Area” y no permitirlo si algún parámetro está fuera de los límites. Otras funcionalidades muy importantes del BMS son: balancear las celdas, calcular la cantidad de carga que le queda (el llamado “State of Charge” o SoC) o almacenar información para posteriores estudios.

El BMS puede ser un sistema analógico o un sistema digital. Los analógicos son capaces de detectar cualquier tipo de parámetro fuera del límite, pero normalmente no son capaces de informar de que parámetro era o por cuanto se ha pasado del límite. Además, no dispone de funcionalidades como la de calcular el SoC o el de almacenar información. Por otra parte, los BMS digitales, que son los más utilizados actualmente tienen las grandes ventajas ya que pueden:

- Ser reconfigurables.
- Ser capaces de procesar la información de las medidas para, por ejemplo, calcular el SoC.

- Comunicarse con los otros sistemas.
- Almacenar información.
- Balancear las celdas.
- Tener más funcionalidades como la detección de errores, ya sean de la propia electrónica como de la batería.

### 3.3.1. Balanceo de celdas

El hecho de tener una batería compuesta por asociaciones de celdas puede provocar que no todas tengan la misma carga. Por ello, es necesario un sistema que se encargue de balancear dichas cargas para equilibrar las cargas.



Figura 2 Ejemplo de un conjunto de celdas no balanceado [9]

La Figura 2 muestra una batería formada por un conjunto de 6 celdas claramente desequilibrados. Sabiendo que son celdas de litio, esta batería no se podría cargar más ya que una de las celdas ya está completamente cargada. En el caso de la descarga se podría hacer, pero, cabe destacar, que una celda esta próxima al voltaje mínimo, por lo que se descargaría muy poco. Una batería de litio está completamente cargada o descargada cuando una de sus celdas llega al límite superior o inferior de voltaje, respectivamente.

Las razones principales de estos desequilibrios son las diferencias de capacidad y de autodescarga ratio de las celdas. Además, si no se balancean las celdas, estos desequilibrios cada vez son más pronunciados. Aunque las celdas sean el mismo modelo, siempre hay pequeñas variaciones en dichas características. A pesar de ello, siempre es recomendable hacer una batería con el mismo modelo de celda de un mismo fabricante.

Para aprovechar al máximo la capacidad de las celdas y, por tanto, de la batería, se tienen que evitar estos desequilibrios. Por lo que, es necesario que el BMS monitorice todos los voltajes de las celdas que componen dicha batería y que tenga un sistema capaz de mantenerlas balanceadas. Hoy en día, los métodos de balanceo más utilizados son el balanceo activo y el balanceo pasivo.

#### 3.3.1.1. Balanceo pasivo

Debido a su simplicidad y su bajo coste, este método es el más utilizado actualmente. Éste se basa en la disipación de energía, es decir, en la descarga de las celdas más cargadas para igualar su carga con las que lo están menos.

Como norma general, este método consiste en la conexión en paralelo de un resistor a las celdas más cargadas para que éstas se descarguen hasta un nivel preestablecido, hasta llegar a la carga de las demás o lo que ocurra antes de las dos opciones anteriores. Como se utilizan resistores para balancear las celdas, estos se pueden seleccionar para hacer que se descarguen más rápidamente o menos según las necesidades del sistema y de las características de las celdas

Este método tiene dos grandes inconvenientes. El primero es la metodología en sí, que se basa en la disipación de energía por lo que, se pierde carga total de la batería para hacer que las celdas estén balanceadas. Usando este sistema, una batería tiene la misma capacidad que la celda con menos capacidad que la componga. El otro inconveniente viene dado también por la metodología, la disipación de energía se hace mediante una resistencia que produce calor, que puede, incluso, ser dañino para las celdas o para el propio BMS.

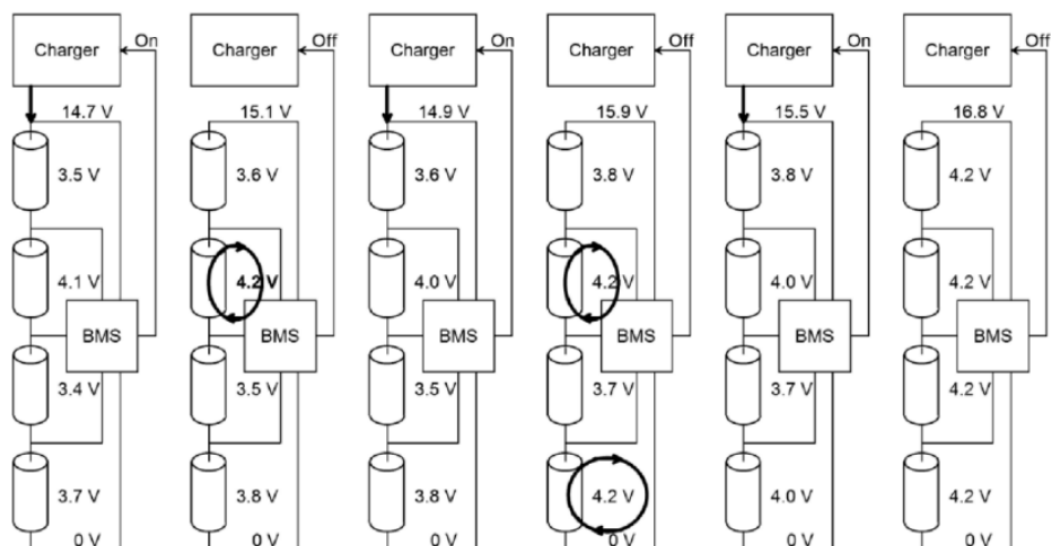


Figura 3 Ejemplo del ciclo de cargado y balanceo pasivo de una batería de cuatro celdas [10]

La **¡Error! No se encuentra el origen de la referencia.** Figura 3 muestra la secuencia de cargado de una batería formada por cuatro celdas de Li-ion. En la primera imagen se ve como las celdas están claramente desequilibradas, dado que hay una con 4,1 V y otra con 3,4 V. La siguiente imagen muestra como la celda que tenía más voltaje llega a su límite superior, por lo que se deja de cargar la batería y se descarga dicha celda con una resistencia en paralelo. Posteriormente, cuando el balanceo ya acabó, se vuelve a cargar la batería hasta que en la siguiente situación se tienen dos celdas que han llegado a su voltaje máximo. En este momento se deja de cargar y estas dos celdas se balancean. Una vez acabado el balanceo se vuelve a cargar la batería hasta que todas las celdas estén al máximo voltaje.

### 3.3.1.2. Balanceo activo

Este método tiene varias formas de hacerlo. La característica principal de este método es que no se basa en la disipación de energía como en el balanceo pasivo, sino que cargan o descargan las celdas entre ellas según convenga.

Existen tres formas de hacer dicho balanceo activo. La primera es hacer que las celdas a medida que vayan llegando a su carga máxima, alimenten a la batería, es decir que la energía sobrante de las celdas más cargadas se utilice para cargar el conjunto de la batería (Figura 4). Otro método es que la batería alimente individualmente a las celdas que tengan menos carga. Estos dos métodos también se pueden combinar para mejorar el rendimiento del balanceo. Por último, existe un método que utiliza solamente las celdas más cargadas para alimentar a las menos cargadas. Básicamente, cualquiera de estos cuatro métodos se basa en utilizar las celdas más cargadas para alimentar a la menos cargadas y reducir los desequilibrios entre ellas.

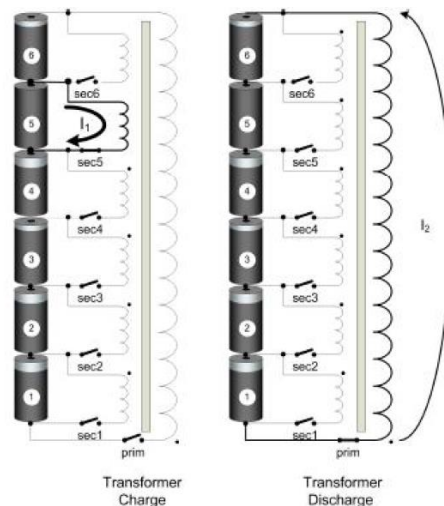


Figura 4 Ejemplo de balanceo activo mediante un convertidor flyback [11]

La Figura 4 muestra una forma de hacer el balanceo activo mediante el primer método mencionado anteriormente. En este sistema la celda que está más cargada se conecta al convertidor flyback para que ésta se descargue mientras carga al transformador. Posteriormente la energía almacenada en el transformador se utiliza para cargar toda la batería.

A pesar de que este método, a simple vista, parezca muy bueno, tiene varias desventajas. La principal es la complejidad comparándolo con el balanceo pasivo. Dicha complejidad provoca que se necesiten más componentes, más cableado y más elementos externos, que a su vez provocan que el coste, peso y espacio aumenten. Por otra parte, la eficiencia del sistema es relativamente baja. Estos inconvenientes hacen que se prefiera optar por un balanceo pasivo a uno activo.

### 3.3.2. Cálculo de parámetros característicos

Existen dos parámetros característicos de las celdas de batería que son el State of Charge (SoC) y el State of Health (SoH). Se utilizan para indicar cuanta batería queda restante y para indicar el estado de vida de la batería, respectivamente.

#### 3.3.2.1. SoC

El SoC muestra en porcentaje la cantidad de carga que le queda disponible a la batería en conjunto. Existen muchos métodos para calcular el SoC, alguno más precisos que otros, pero en lo que respecta al BMS, en general, se utilizan métodos de cálculo simples debido a la limitación de procesamiento de excesivos datos de los microcontroladores. Por ese motivo se dice que el SoC no es un cálculo exacto sino una estimación. Además, cabe destacar, que los microcontroladores suelen hacer varias tareas por lo que no pueden destinar toda su capacidad al cálculo de este parámetro. Los métodos más comunes para la estimación del SoC de una batería son el OCV y el Coulomb Countign.

Normalmente, los parámetros que se suelen utilizar para estimar el SoC son el voltaje, la corriente y la temperatura. El voltaje porque hay una relación directa entre la capacidad de la celda/batería y el voltaje en circuito abierto de esta, conocido como “open circuit voltage” o OCV. La corriente porque la corriente de la celda/batería es proporcional a la capacidad que pierde o gana. Por último, la temperatura porque se ha demostrado que la capacidad de una celda/batería varía con la temperatura (Figura 5).

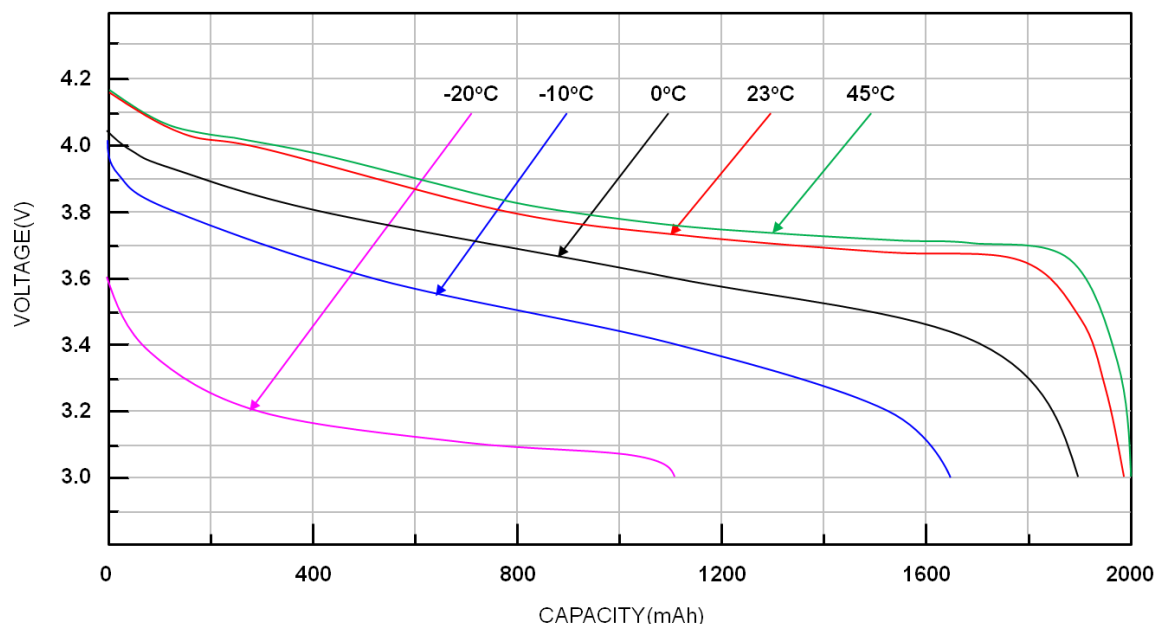


Figura 5 Ejemplo de la variación de capacidad total de una celda Li-ion con la temperatura [12]

El primer método, el ya nombrado OCV, consiste en determinar la capacidad según el voltaje en circuito abierto de la batería. Esta relación se puede obtener haciendo una serie de test a las celdas. Las ventajas que tiene es que es fácil de llevar a cabo, pero, por otra parte, requiere medidas de voltaje muy precisas y solo se puede hacer cuando la batería está en circuito abierto. Para este último inconveniente existen varias soluciones. Una es la de hacer el test que relaciona el voltaje con la capacidad a diferentes corrientes de carga/descarga (Figura 6), y mediante las medidas de voltaje e intensidad se elige una relación u otra. Otra posible solución es calcular el voltaje que tendría en circuito abierto, a partir de las medidas de corriente, voltaje y de la modelización de la celda/batería. Además, si se quiere tener en cuenta también el efecto de la temperatura, se tendría que hacer una gran cantidad de test.

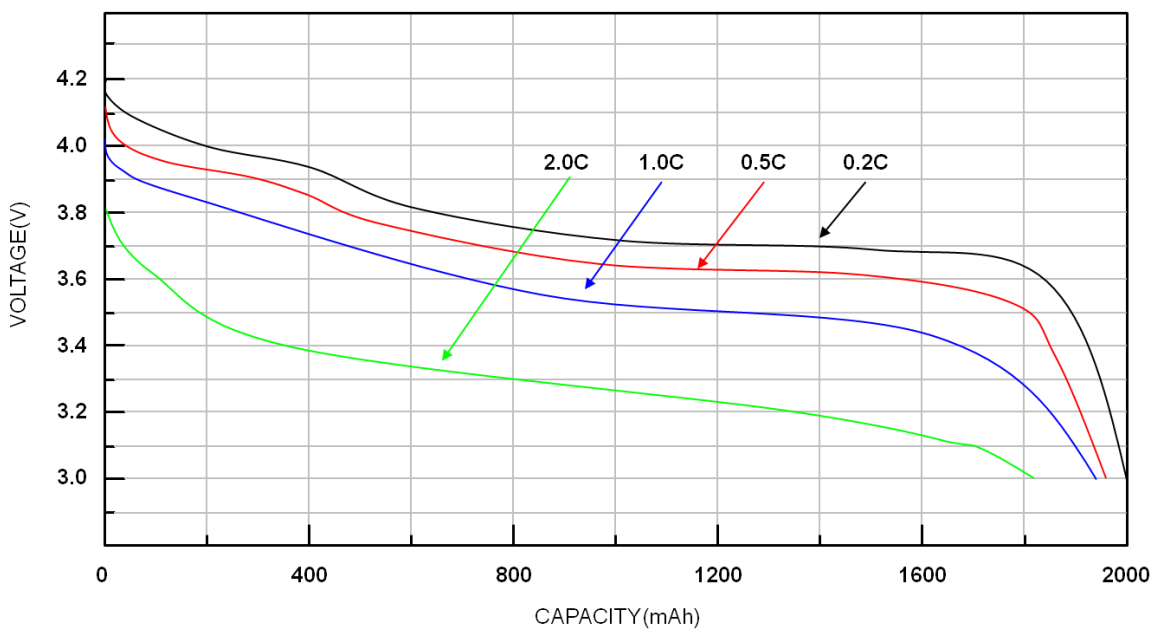


Figura 6 Ejemplo de la variación de voltaje y capacidad total según la corriente de descarga [12]

Otro método es el Coulomb Counting que se basa en la integración de la intensidad con respecto al tiempo para saber la capacidad que gana o pierde la celda (Ec.1) [13]. Esta forma de estimación del SoC es una de las más usadas actualmente debido a sus grandes resultados y su relativa sencillez. Los inconvenientes que tiene son, por una parte, que depende del estado inicial y que los errores se acumulan al basarse en integraciones, y, por otra parte, que requiere medidas de corriente muy precisas y que no tiene en cuenta la autodescarga de la celda/batería. Para solucionar los errores de integración lo que se suele hacer es estimar el estado inicial a partir de otros métodos, y hacer periódicamente recalibraciones para eliminar los posibles errores que se arrastran.

$$SOC(t) = SOC(t_0) + \frac{\int_{t_0}^t i * dt}{Ah} * 100 \quad (\text{Ec. 1})$$

### 3.3.2.2. SoH

El State of Health indica, como se ha dicho anteriormente, el estado de la vida de la celda/batería. Es decir, indica cuanta capacidad total ha perdido respecto a la capacidad máxima que tenía. Se suele indicar en porcentaje (Ec. 2).

$$SOH(t) = \frac{Q_M}{Q_f} * 100 \quad (\text{Ec. 2})$$

Donde  $Q_M$  (en Ah) es la capacidad actual de la batería y  $Q_f$  (en Ah) es la capacidad máxima que tenía la batería.

Como indica la Ec.2 se indica el porcentaje útil de capacidad en función de la capacidad máxima, es decir, el porcentaje de la capacidad total máxima que queda disponible. Por ejemplo, indicar que a la batería de 4000 mAh le queda un 87,5% de la capacidad inicial quiere decir que su capacidad actual es de 3500 mAh.

Existen varios métodos para calcular el SoH. El más utilizado actualmente se basa en un test de descarga. En dicho test la batería se descarga, se calcula la capacidad extraída de la batería a partir del Coulomb Counting mencionado anteriormente, y finalmente, se calcula el SoH.

### 3.3.3. Topología

Dado que las baterías de los coches suelen estar compuestas por un gran número de celdas y que el BMS es un sistema electrónico, existen varias posibles topologías para este. Las topologías más comunes son: la centralizada, la descentralizada y el modular. Después existen combinaciones entre ellas.

#### 3.3.3.1. Centralizado

Esta topología se basa en un sistema que monitoriza las celdas desde una placa central. Esta topología se utiliza en los BMS analógicos o en los BMS digitales de batería relativamente pequeñas de menos de 20 celdas.

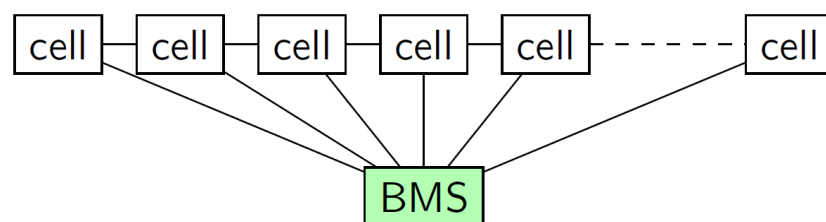


Figura 7 BMS centralizado [9]

Normalmente son sistemas más simples y más baratos de diseñar. En el caso de que sea un sistema digital solo se tiene que reprogramar un microcontrolador para cambiar lo que se quiera del software. El inconveniente más grande es que es una topología pensada para baterías pequeñas.

### 3.3.3.2. Descentralizado

Esta topología es totalmente opuesta a la centralizada. En este caso el BMS está dividido en subsistemas, donde cada uno de estos monitoriza una o varias celdas. Además, requiere que sea digital ya que se necesita que dichos subsistemas se comuniquen entre sí.

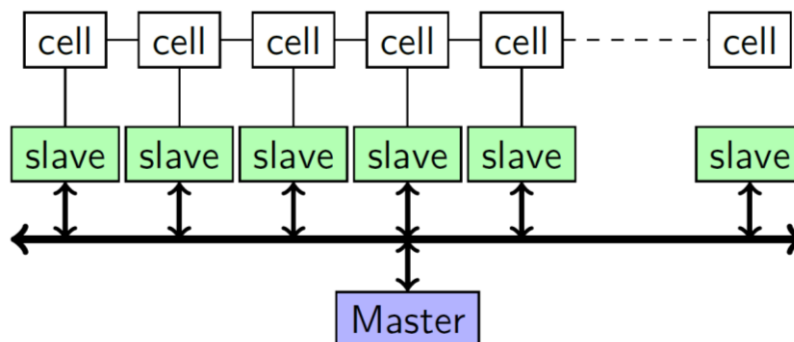


Figura 8 BMS descentralizado [9]

Los *slaves* se encargan de monitorizar el voltaje y la temperatura de las celdas y la master se encarga de recopilar la información y procesarla para, por ejemplo, calcular el SoC. La comunicación entre los *slaves* y la master puede hacerse siguiendo varios protocolos, como el CAN, I2C, SPI entre otros.

La ventaja principal de esta topología es la versatilidad que tiene. Se puede añadir o quitar celdas con sus respectivos *slaves* de la batería con facilidad sin que afecten al sistema general. Otro punto a favor es que no hay problemas para las baterías con gran cantidad de celdas. Como desventaja se puede decir que es un sistema más complicado de diseñar, porque se tiene que tener en cuenta el protocolo de comunicación y las medidas necesarias para hacerlo funcionar de manera correcta. Además, una actualización de software requiere ir *slave* por *slave* reprogramando.

### 3.3.3.3. Modular

Un BMS modular está formado por diferentes módulos independientes. Esta topología es una combinación de las dos anteriores. Cada módulo monitoriza un grupo de celdas y se comunica con los demás módulos para transmitir la información obtenida.



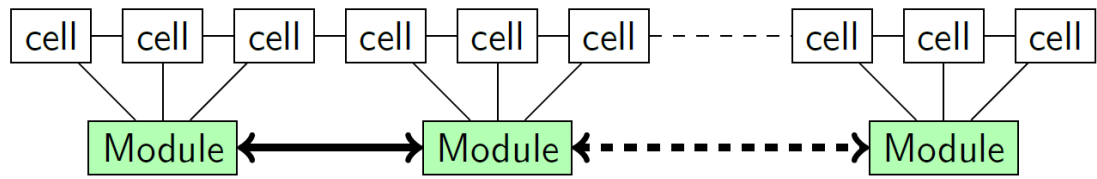


Figura 9 BMS modular [9]

Normalmente uno de los módulos, además de sus funciones habituales, también se encarga de recopilar la información y procesarla. Comparte gran parte de las ventajas y desventajas de la topología descentralizada. Aun así, tiene puntos a favor y en contra propios. No necesita una master ya que cada módulo es independiente, pero si se quiere aumentar o disminuir el número de celdas de la batería se tiene que hacer en un múltiplo de la cantidad de celdas que controla cada módulo.



## 4. Diseño del BMS

En este trabajo se ha optado por el diseño de un BMS digital, dado que es el más extendido y del cual ya se tenía alguna referencia [14] ,[15]. El hecho de que sea digital comporta que el diseño se divida en dos fases, el diseño hardware y el diseño software. Otro motivo para esta decisión es el hecho que muchos fabricantes ofrecen circuitos integrados con funciones muy específicas de control de baterías, lo que facilita el diseño hardware, pero complica el diseño software. También se decidió que el balanceo sería pasivo debido a su sencillez. Por último, la topología de este BMS es centralizada, pero es posible adaptarlo para ser modular con pequeños cambios. Se eligió esta topología debido a la batería que se utiliza, que solo tiene 12 celdas. Por lo que con un solo microcontrolador es suficiente.

Aunque la idea final de este proyecto sea que este BMS se adapte para ser utiliza en la batería de un Formula Student, la realización se hace en un batería de 12 celdas. Dicha batería está proporcionada por el equipo de la universidad, e-Tech Racing. Tiene una configuración de 12s1p (12 celdas en serie sin ningún paralelo) y está compuesta por celdas de  $\text{LiFePO}_4$ .

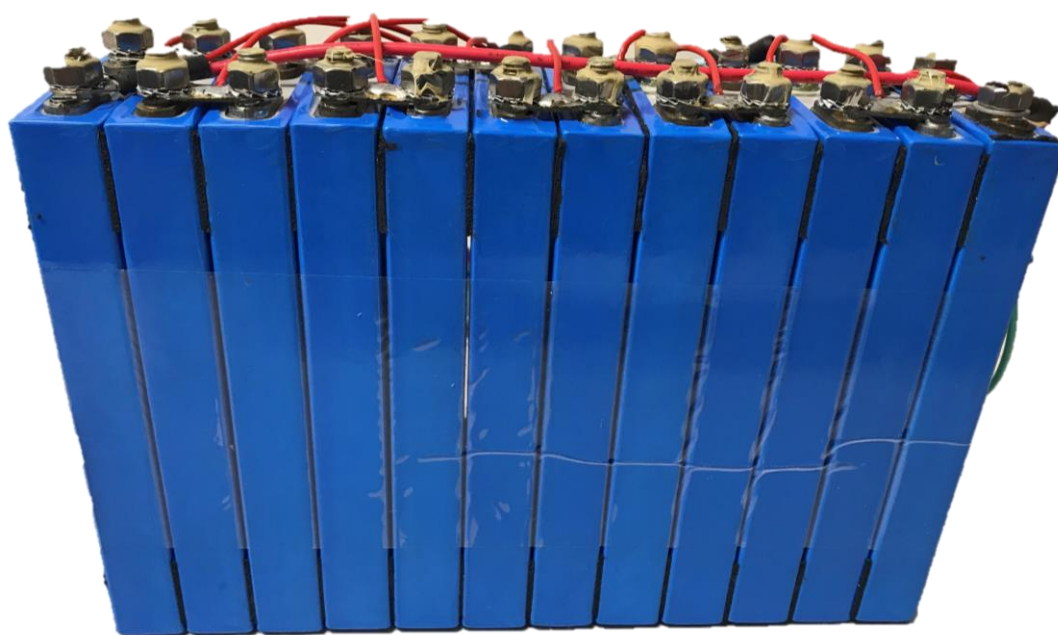
### 4.1. Diseño del hardware

El diseño de hardware se ha enfocado de la siguiente manera. Primero se ha definido como obtener las medidas necesarias y las comunicaciones para la realización del BMS. Posteriormente se ha elegido el microcontrolador que sea capaz de controlar todos los periféricos y sea capaz de realizar los cálculos pertinentes. En el Anexo A se muestran imágenes más detalladas de los esquemáticos.

#### 4.1.1. Batería

La batería no es un elemento diseñado, sino que es el elemento a controlar, y por tanto se debe conocer. En este apartado se describen el tipo de batería que es, sus características y porque se ha elegido.

La batería (modelo LFP-4810S 48V/10AH) fue fabricada para una bicicleta eléctrica por Golden Motors[16]. Inicialmente estaba compuesta por 16 celdas de  $\text{LiFePO}_4$  asociadas en serie y tenía su propio BMS. Para la aplicación de este trabajo sólo interesaba la batería por lo que el BMS propio que tenía se desconectó y, además, se decidió quitar 4 celdas para hacer una batería final de 12 celdas de  $\text{LiFePO}_4$  en serie (configuración de 12s1p) (Figura 10) y utilizar estas 4 celdas sueltas, para testearlas y obtener datos importantes para la estimación del SoC. Las características de la batería final están reflejadas en la Tabla 2.



*Figura 10 Batería final que se usará para probar el BMS.*

*Tabla 2 Características de la batería*

Número de celdas	12
Configuración	12s1p
Capacidad de las celdas que componen la batería	10 Ah
Voltaje máximo	43,2 V
Voltaje mínimo	30 V
Voltaje nominal	38,4 V
Energía total	432 kWh
Máxima corriente de descarga	35 A
Máxima corriente de descarga continua	20 A
Máxima corriente de carga	4 A

Por último, se ha elegido hacer el BMS de esta batería por temas económicos. No había presupuesto como para comprar celdas para hacer una batería a medida. Además, el equipo de Formula Student

de la universidad, e-Tech Racing tenía la que se ha descrito anteriormente y estaba dispuesto a dejarla para realizar este trabajo. Por esos dos motivos la elección de la batería fue la descrita en este apartado.

#### 4.1.2. Obtención de voltajes y balanceo

Una de las funciones principales del BMS es la monitorización de todos los voltajes y el balanceo de celdas en caso de cierta diferencia de tensión entre ellas. Podrían ser diseños de hardware diferentes, pero, debido a la solución adoptada, se explican en el mismo apartado.

El problema principal de la obtención de voltajes es que puede haber voltajes bastantes elevados como para leerlos con un solo ADC (*analog to digital converter*). Por ese mismo motivo, se pensó en adaptar cada señal antes de entrar al ADC (que podía ser externo al microcontrolador o interno a él) mediante un amplificador operacional diferencial. El problema de este método es que para cada celda necesitarías su propio amplificador, lo que haría el sistema muy grande al final. Posteriormente, se pensó en usar solo un amplificador y multiplexar las señales de las celdas anteriormente. De esta manera se reducían el número de amplificadores, pero se añadían uno o dos multiplexores. El problema era que seguía siendo un hardware bastante grande y, además, se tenían que buscar multiplexores que fuesen capaces de aguantar entradas y salidas de más de 40 V.

Esa posible solución tampoco tenía en cuenta el balanceo de las celdas. Por eso la posibilidad de seleccionar un circuito integrado que haga la función de obtener los voltajes empezó a ser más convincente. El inconveniente de esta posibilidad era que, como son muy específicos, suelen ser tan pequeños que la soldadura a mano es extremadamente complicada. A pesar de ello se decidió seleccionar uno de ellos, soldarlo y probarlo. Para acotar la búsqueda de dicho integrado se empezó por seleccionar el fabricante priorizando aquellos que proporcionen muestras de sus productos, debido a un presupuesto limitado. Dentro de estas posibilidades, se tenía que tener en cuenta que se tenían que medir 12 voltajes y, si se pudiese, hacer el balanceo. De entre todos los fabricantes se seleccionó Linear Technology por varios motivos: proporciona muestras gratis a estudiantes, tiene documentos adicionales al datasheet para simular el componente o para ayudar a utilizarlo, tiene placas hechas con dichos componentes con el hardware necesario para probarlo, porque se tenía el contacto de un distribuidor que podía resolver cualquier tipo de dudas y, porque es la empresa más puntera en este sector específico de control de baterías.

Dentro de los modelos que tiene, se optó por el LTC6811-2 [17] ya que, aparte de ser de los pocos modelos que permitía muestras, es el más novedoso. Permite medidas de voltaje con errores inferiores a 1,5 mV y permite una configuración de balanceo pasivo. Por otra parte, se controla mediante un microcontrolador externo, y existen varios protocolos de comunicación para comunicarlos: Daisy Chain, SPI o I2C. Debido a que se tenían pocas nociones sobre el protocolo Daisy Chain y que para usar el I2C se tenía que añadir extra hardware, se decidió hacer dicho control y comunicación mediante SPI.

Por ese motivo se seleccionó el LTC6811-2 (preparado especialmente para esta comunicación) en vez de su hermano, el LTC6811-1 (preparado para la comunicación Daisy Chain).

Aunque se tuviese seleccionado el circuito integrado a utilizar, seguía existiendo el problema de la soldadura, que era muy complicada para hacerla a mano. Finalmente, y gracias a la ayuda del distribuidor mencionado anteriormente, se consiguió lo que en Linear Technology llaman demoboard. Esta demoboard consiste en una PCB (printed circuit board) que tiene ya soldados el circuito integrado y el hardware necesario para el correcto funcionamiento del LTC6811-2, por lo que la solución final para hacer la obtención de voltajes y el balanceo fue la demoboard DC2260A (Figura 11) que contiene el circuito integrado que se quería utilizar, el LTC6811-2.

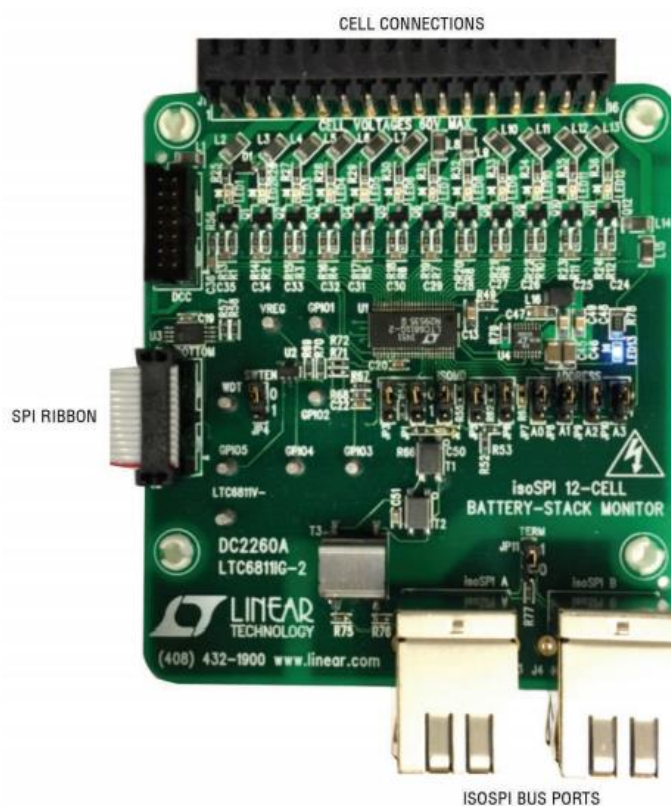


Figura 11 Demoboard DC2260A

El esquemático de la placa DC22660A está representado en la Figura 12. Aunque el diseño ya estuviese hecho, es recomendable conocerlo y entenderlo por dos motivos principales: el primero es que facilita la programación y el segundo es que, si en el futuro se hará el BMS completo, seguramente no se tenga el presupuesto necesario para comprar las demoboards necesarias y, por tanto, se tendrán que diseñar.

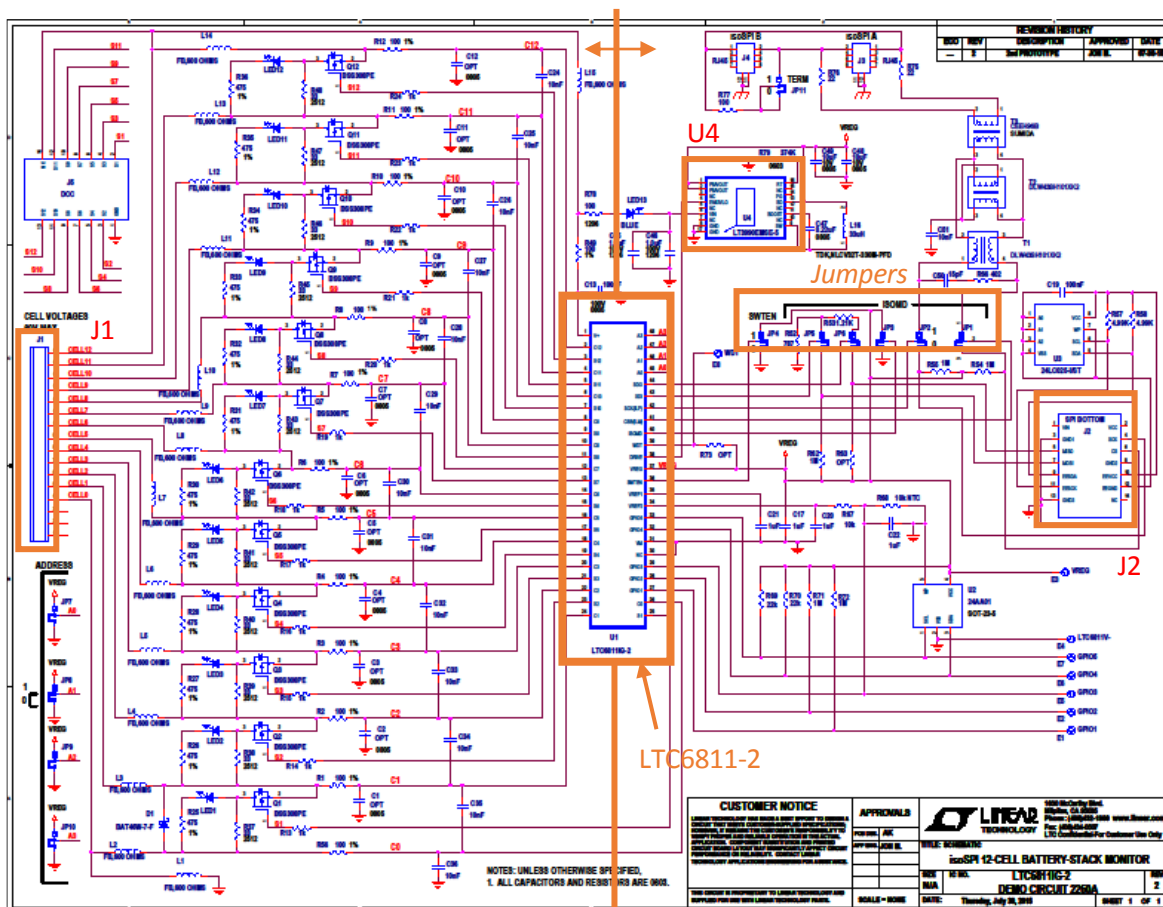


Figura 12 Esquemático de la Demoboard (DC2260A) de Linear Technology que contiene el LTC6811-2 con el hardware necesario para que funcione correctamente [18]

En el centro de la Figura 12 se puede observar el circuito integrado LTC6811-2. La parte de la izquierda del integrado corresponde a la obtención de voltajes y el balanceo y la parte de la derecha corresponde a las comunicaciones. En el Anexo A se han adjuntado imágenes del esquemático en partes para poder apreciar mejor los detalles.

El conector J1 es donde se conectan las celdas. La señal de este conector esta filtrada a través de una ferrita y de un filtro pasa bajos, que, además de ayudar a estabilizar la señal reduciendo posible ruido, limita la corriente que puede entrar en el integrado. Estas señales filtradas se conectan a los pines Cx (donde x es el número de la celda). Cabe destacar que esta estructura de filtrado también se utiliza para alimentar el integrado a través de los pines  $V^+$  y  $V^-$ .

La estructura de balanceo está compuesta por una resistencia de balanceo que es la que se encarga de disipar la energía; por un LED junto a una resistencia para limitar la corriente que pasa por el LED que sirven para indicar cuando se está balanceando una determinada celda; y, finalmente por un PMOS

cuya *gate* está conectada a los pines *Sx* a través de una resistencia para hacer posible el control de dichos MOSFETs por el LTC6811-2.

Para alimentar la electrónica de control interna del circuito integrado se utiliza un DC/DC (el componente U4) que a partir de la batería genera 5V. Un detalle es que esta alimentación de 5V no es constante, por lo que la entrada del DC/DC tiene un LED que sirve para indicar cuando está funcionando y cuando no. Esto lo controla el propio LTC6811-2 a través del *enable* del convertidor.

Los *jumpers* JP1, JP2, JP3, JP4, JP5 y JP6 están puestos en la posición 0 para configurar la comunicación por el protocolo SPI o I2C, en vez de usar el isoSPI. Estos *jumpers* simplemente conectan directamente los cuatro pines del SPI del LTC6811-2 al conector J2, que también tiene los pines del I2C. Cabe destacar que los las señales de MISO y MOSI del SPI tienen una resistencia de *pull-up* para mejorar la comunicación. Aunque se utilice solamente la comunicación SPI, hay que destacar que las señales de I2C se generan a través de los circuitos integrados U3 y U2 y que los conectores J4 y J5 junto a los tres transformadores se podrían utilizar para la comunicación isoSPI. Por último, en el tema de las comunicaciones se tiene que destacar que como SPI es un protocolo que tiene direccionamiento, se tiene que definir la dirección del chip. Esto se hace mediante los *jumpers* JP7, JP8, JP9 y JP10 que se han puesto en la posición 0, por lo que la dirección del LTC6811-2 es 0000.

Finalmente, se tiene que comentar dos cosas más. La primera es que hay puntos de testeo del GND del circuito, de los pines GPIO (que no se utilizan en este caso) del circuito integrado, de la alimentación de 5V y del pin WDT. La segunda es que el J2 es un conector cuyos pines están conectados a las salidas *Sx* que sirve para comprobar con un multímetro o un microcontrolador externo el estado de dichas salidas.

#### 4.1.3. Medición de temperatura

Para medir la temperatura existen varias maneras de hacerlas. Se pueden hacer con sensores ya prefabricados, el problema es que suelen ocupar bastante espacio. Otras posibilidades descartadas por su precio eran los termopares y las a RTD (resistance temperatura detector). Finalmente, se optó por el uso de termistores debido a su simplicidad, por su reducido coste y por la experiencia previa con este sensor.

Dentro de los termistores hay dos tipos: las PTC (positive temperatura coefficient) y las NTC (negative temperatura coefficient). Cuando la temperatura aumenta, las primeras incrementan su resistencia mientras que las otras la disminuyen. Dado que las PTCs son más inestables y menos repetitivas, se eligió usar NTCs. La relación entre la temperatura y la resistencia en un termistor viene dada por la ecuación, Ec. 3.



$$R(T) = R_{REF} * e^{\beta * (\frac{1}{T} - \frac{1}{T_{REF}})} \quad (\text{Ec. 3})$$

Donde  $R(T)$  es la resistencia que tendrá en una determinada temperatura  $T$ ,  $R_{REF}$  es la resistencia que tiene dicha NTC en  $T_{REF}$ ,  $\beta$  es la temperatura característica,  $T$  es la temperatura que tiene la NTC y  $T_{REF}$  es la temperatura de referencia. Las unidades son ohmios ( $\Omega$ ) para las resistencias y Kelvin (K) para las temperaturas. Los valores de  $R_{REF}$ ,  $\beta$  y  $T_{REF}$  te lo proporciona el fabricante. En este caso la NTC seleccionada es NXRS15XV103FA1B040 de Murata debido a que son las proporcionadas por e-Tech Racing. Los datos de esta NTC son los siguientes:  $R_{REF}$  es 10 k $\Omega$ ,  $T_{REF}$  es 25 °C y  $\beta$  es 3971 K.

Sabiendo la resistencia de la NTC, se puede obtener la temperatura. Por tanto, lo que se hizo fue conectarla a un divisor de tensión, de manera que, sabiendo el voltaje de alimentación, el valor de la otra resistencia y la caída de tensión en el punto medio de ellas, se puede conocer la resistencia de la NTC, y, consecuentemente, la temperatura. En este punto, surge otro problema, el hecho de que el comportamiento de la resistencia de la NTC sea exponencial provoca, que el procesamiento posterior que hará el microcontrolador ocupara mucha memoria de este. Para facilitar dicho procesamiento, evitando las funciones exponenciales se optó por linealizar dicho comportamiento conectando una resistencia en paralelo a la NTC. Esta opción optimiza el programa del microcontrolador, pero reduce la exactitud de la medida. Como normalmente, las baterías de los coches de Formula Student, están refrigeradas, estas no trabajarían nunca fuera de su temperatura máxima. Asumiendo esta condición, es aceptable tener cierto error en la medida si se puede simplificar el hardware o el software y reducir costes ya que situaciones donde la temperatura sea excesiva no tendrían que darse.

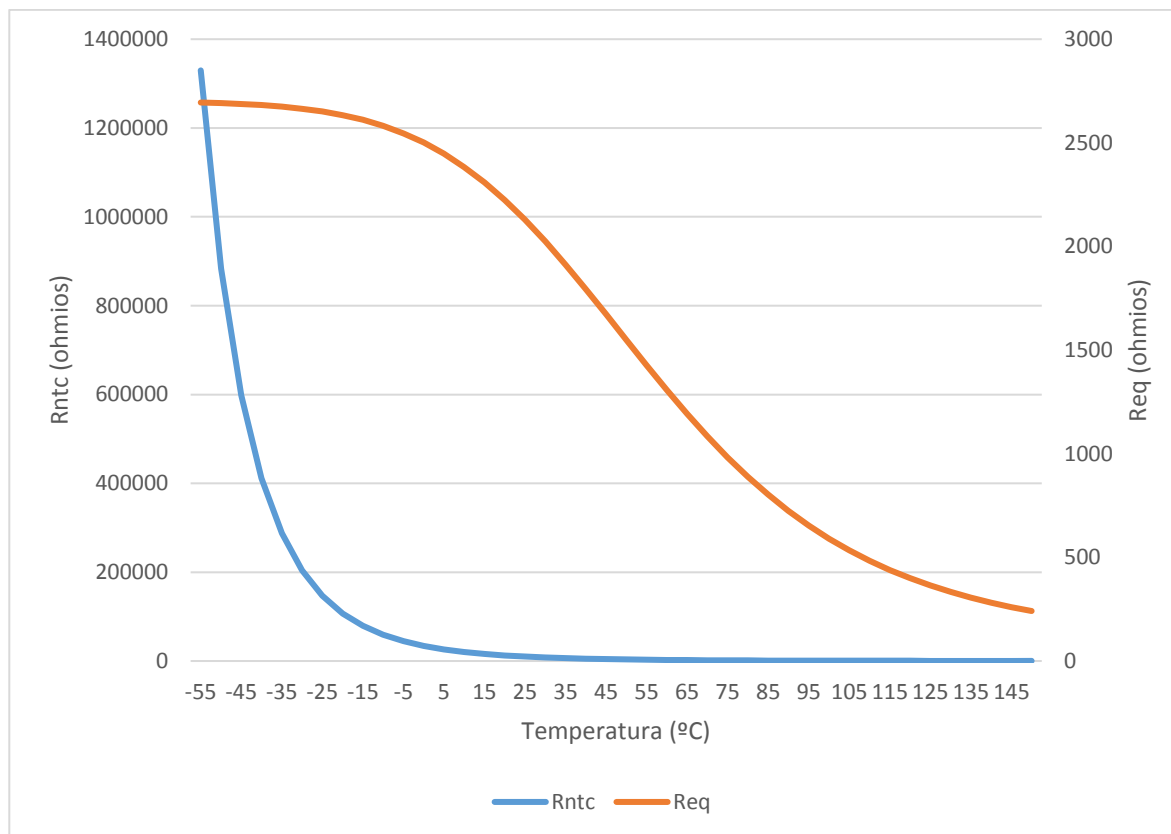


Figura 13 Comparación del comportamiento de la resistencia de una NTC ( $R_{ntc}$ ) y de una NTC linealizada con una resistencia en paralelo ( $R_{eq}$ ).

Como se observa en la Figura 13, si se linealiza la NTC, la resistencia equivalente tiene un rango que se puede aproximar a una recta lineal. Eligiendo de manera adecuada la resistencia que se pone en paralelo a la NTC se puede conseguir dicho rango lineal en un rango de temperaturas u otro. Debido a que la normativa de Formula Student especifica que la batería no puede trabajar a más de 60°C, se ha definido dicho rango alrededor de esta temperatura. La ecuación, Ec.4 [19], se ha utilizado para calcular la resistencia paralela necesaria para el rango querido.

$$R_p = R_{TC} * \frac{\beta - 2 * T_c}{\beta + 2 * T_c} \quad (\text{Ec. 4})$$

$R_{TC}$  es la resistencia (en ohmios) de la NTC en la temperatura  $T_c$  (en K) donde se requiere la zona lineal,  $\beta$  (en K) es la temperatura característica de la NTC. Para este cálculo se ha definido  $T_c$  como 60 °C o 333 K,  $\beta$  como 3971 K y  $R_{TC}$  como 2534,54 Ω. El resultado obtenido es 2534,54 Ω, por lo que, como valor normalizado, finalmente se ha elegido 2,7 kΩ.

Finalmente, hay que hacer la aproximación a una recta. Para ello se utiliza la forma de hacer una recta mediante dos puntos. Estos se definen como  $T=25^{\circ}\text{C}$  y  $T=60^{\circ}\text{C}$ . En estos puntos, el error de resistencia será nulo teóricamente, por ese motivo se ha seleccionado  $T=60^{\circ}\text{C}$ . El otro punto se seleccionó porque

cuanto más pequeño sea el rango menos errores obtendrás al final. Por otra parte, se han permitido errores de medida considerables el rango inferior a 25°C, ya que se considera que la batería siempre estaría por encima de los 0°C por la situación geográfica, por lo que no tendría peligro por bajas temperaturas. De esta manera la ecuación lineal resultante es la siguiente:

$$R(T) = -23,48 * T + 2716,13 \quad (\text{Ec. 5})$$

Donde  $R(T)$  (en ohmios) es la resistencia aproximada en la temperatura  $T$  (en °C). Las unidades de las dos constantes, -24.48 y 2716,13 son  $\Omega/^\circ\text{C}$  y  $\Omega$ , respectivamente.

La resistencia serie del divisor de tensión se seleccionó de forma iterativa teniendo en cuenta los voltajes reales resultantes. Si se seleccionaba una resistencia muy grande la variación de voltaje del divisor a diferentes temperaturas es mínima, y si es muy pequeña pasa lo contrario. Finalmente, se optó por una resistencia de 4,7 k $\Omega$ .

Posteriormente, dicho voltaje se tiene que adaptar para aprovechar al máximo el rango de funcionamiento del ADC (de 0V a 5V) para obtener una mayor resolución. Para adaptarlo se decidió usar un amplificador operacional *railtorail* (LTC6201 [20]) con una configuración de restador (Figura 14) alimentado a 5V.

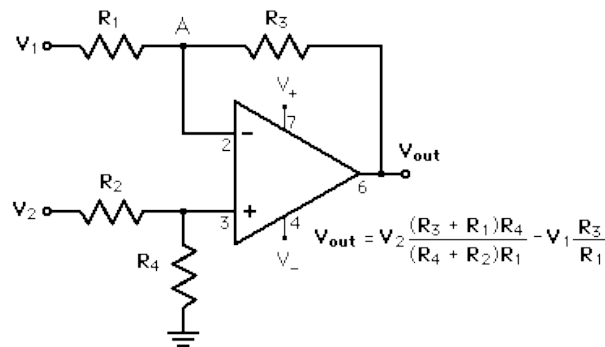


Figura 14 Esquemático del amplificador operacional diferencial y ecuación de salida de este

Primero, se definió a que temperatura se querían los 0 V o, dicho de otra forma, que voltaje real se quería adaptar a 0 V. Se decidió que esa temperatura fuera 0°C considerando que la batería nunca estaría por debajo de dicha temperatura. Posteriormente se eligió a que temperatura le corresponderían los 5 V, en este caso fue 70 °C. Con los voltajes reales del divisor a estas temperaturas se definió la función (Ec.6) que debería seguir la etapa adaptadora.

$$V_{out} = -6,25 * V_{in} + 10,85 \quad (\text{Ec. 6})$$

Donde  $V_{in}$ ,  $V_{out}$  y la constante 10,85 están en voltios y la constante -6,25 está en voltios/voltios.  $V_{in}$  sería la salida de un seguidor de tensión (hecho con el mismo LTC6201) del voltaje obtenido en el divisor de tensión, de esta manera se evita que las posibles resistencias de la etapa acondicionadora interfieran en los cálculos anteriores. Haciendo la comparación con la ecuación del amplificador operacional  $V_{in}$  correspondería con  $V_1$  y los demás parámetros se tienen que calcular. Fijando  $V_2$  en 2,5 V mediante una referencia de tensión (LTC1019[21]),  $R_4$  en 27 k $\Omega$  y  $R_3$  en 20 k $\Omega$ , se necesitaría que  $R_1$  y  $R_2$  fueran 3198,51  $\Omega$  y 16889,62  $\Omega$ , respectivamente. Finalmente, eligiendo valores comerciales de  $R_1$  y  $R_2$  de 3,3 k $\Omega$  y 16 k $\Omega$ , respectivamente, la ecuación (Ec.7) resultante es la siguiente.

$$V_{out} = -6,06 * V_{in} + 11,08 \quad (\text{Ec. 7})$$

Donde  $V_{out}$ ,  $V_{in}$  y la constante 11,08 están en voltios y la constante -6,06 está en voltios/voltios. Se aproxima bastante a la ecuación esperada. De esta manera en 0 °C se obtiene un voltaje final real de 0,55 V y en 70 °C se obtiene 5,4 V. Dentro del rango de temperatura de 25 °C a 60 °C que se consideró para hacer la aproximación, el error máximo teórico es inferior a 1 °C, aunque también se tiene que decir que desde 25 °C a 0 °C (rango no importante) el error asciende hasta 9,15 °C.

El microcontrolador tendrá que hacer el camino inverso para obtener el valor de la temperatura, ya que lo que recibe es el valor del ADC interno de  $V_{out}$ . Mediante este valor se podrá obtener  $V_{in}$ , mediante  $V_{in}$  se obtendrá  $R(T)$  y mediante  $R(T)$  y la recta aproximada se obtendrá la temperatura  $T$ .

Cabe destacar que dicho hardware no se ha repetido para cada NTC, sino que se han multiplexado las NTC con un solo montaje linealizador y adaptador para optimizar el espacio de la PCB. La Figura 15 muestra el diseño final que se ha explicado en este apartado.

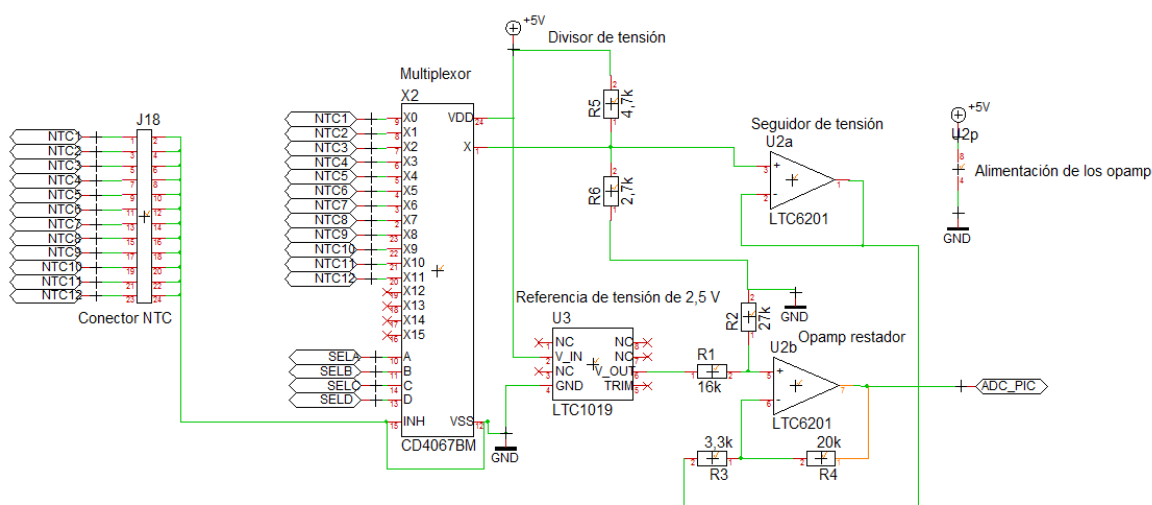


Figura 15 Esquemático final de la obtención de temperatura

#### 4.1.4. Medición de corriente

La medición de corriente se puede hacer por dos métodos. El primero es mediante una resistencia *shunt*, esta tiene un valor de resistencia lo más pequeño posible y se pone en serie con un polo de la batería. Se mide la caída de tensión en bornes de dicha resistencia, por lo que sabiendo dicho voltaje y el valor de la resistencia se puede conocer la corriente a partir de la ley de Ohm. El segundo método es mediante un sensor hall. Este sensor lo que realmente hace es medir el campo magnético que es proporcional a la corriente que lo crea. Son sensores cuya salida suele ser un voltaje directamente proporcional a la corriente que miden.

Debido a que durante las temporadas pasadas temporadas las mediciones de corriente de la batería de alta tensión del coche se han hecho mediante sensores hall se ha elegido esta medición de la misma manera. La primera idea era usar uno de los sensores hall que podía proporcionar el equipo, el HASS-50S (Figura 16) de LEM. Este sensor se alimentaba a 5V y daba la medida de corriente en voltaje, donde los 0 A eran 2,5 V y según la dirección de la corriente este voltaje aumentaba o disminuía con una sensibilidad de 12,5 mV/A. Dicha sensibilidad era muy grande por lo que al hacer medidas a pequeñas intensidades no se conseguía la precisión deseada. Como se explicarán en apartados posteriores, se requería que la medida de corriente fuese mínimamente exacta. Por este motivo y el hecho que para este proyecto no haría falta tener un rango de medida excesivamente elevado, se decidió descartar la opción de usar este modelo.



Figura 16 Imagen del sensor hall de corriente HASS-50S de LEM

Se empezó a buscar modelos de corriente de efecto hall parecidos al HASS-50S. Todos los encontrados tenían alguna característica que no acababa de encajar, por ejemplo, tenían una alimentación de  $\pm 12$ , eran sensores SMD, o tenían un rango de medida demasiado grande o demasiado pequeño. Finalmente, se eligió el CAS 6-NP (Figura 17) del mismo fabricante, LEM. Este modelo comparte ciertas características como la alimentación de 5V y el tipo de salida: un voltaje directamente proporcional a la corriente que mide. Es capaz de medir de +20 A a -20 A, por lo que la sensibilidad es bastante menor: 104mV/A. El inconveniente que tenía es que está pensado para PCB, no para estar en el aire por lo que no tiene un conector como el anterior, ni para el camino de la corriente a medir ni para la alimentación y la salida. Este problema se solucionó soldando los cables directamente a los terminales del sensor.



Figura 17 Imagen del sensor hall de corriente CAS 6-NP de LEM

#### 4.1.5. Can hardware

Dado que los microcontroladores son componentes digitales y el protocolo de comunicación CAN se basa en voltajes diferenciales, se necesita un componente intermedio que entienda las dos partes y haga de buffer. Por una parte, debe ser capaz de recibir y enviar información digital para comunicarse con el microcontrolador. Por otra parte, debe ser capaz de enviar y recibir información por la línea de CAN y de gestionarla. Este componente se llama *CAN transceiver*.

De entre todos los fabricantes que hacen dicho componente se optó por mirar en Microchip, ya que, aparte de dar muestras a estudiantes, es el que hace el modelo que utiliza como norma general los sistemas electrónicos del coche de e-Tech Racing. Finalmente, el modelo seleccionado es el MCP2551 porque es el que más se conoce dentro del equipo y porque es el modelo con el que se hicieron pruebas con este protocolo.

#### 4.1.6. Microcontrolador

El microcontrolador elegido es de Microchip como el *CAN transceiver*. Ésta decisión viene dada por el hecho de que era el microcontrolador disponible para hacer las pruebas previas al diseño de la PCB (pruebas de comunicación CAN para el coche, de comunicación SPI para controlar el circuito integrado de Linear y de medición de temperatura). Además, tiene unas características que cubren con holgura las necesidades del proyecto: tiene mucha memoria de programación como para hacer muchas operaciones matemáticas, tiene entradas y salidas tanto digitales como analógicas y permite varios tipos de comunicación. El modelo es el PIC18F2685.

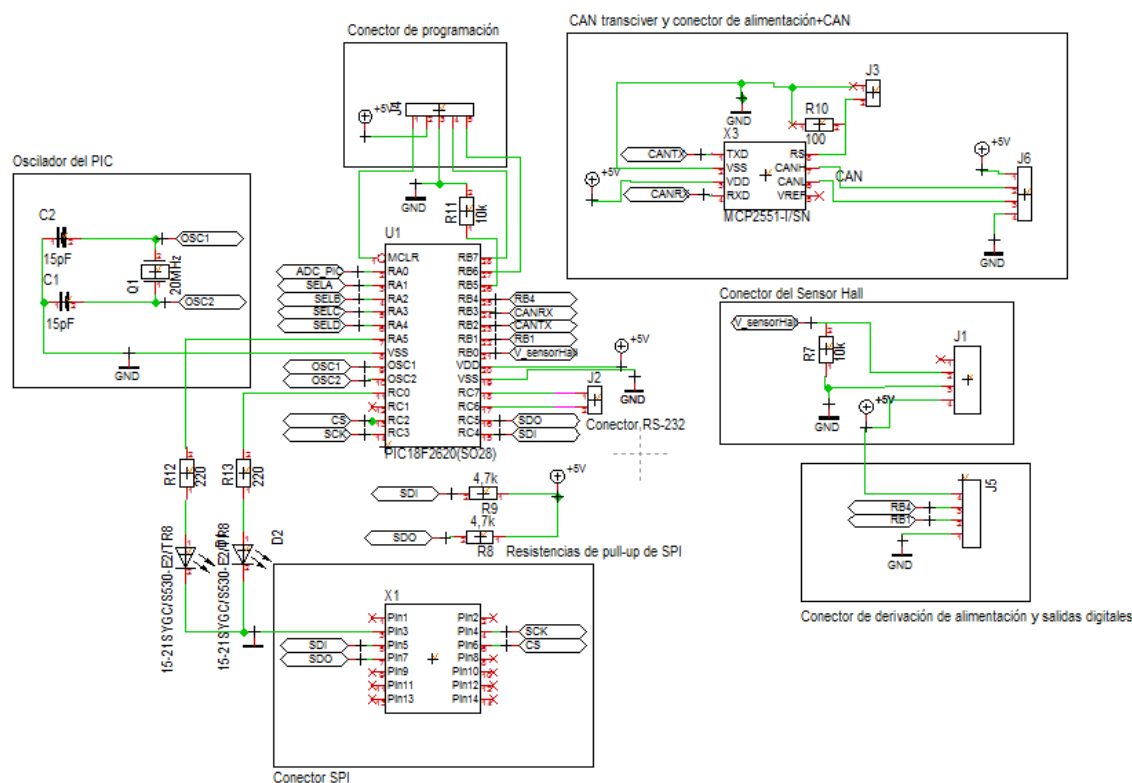


Figura 18 Esquemático del conexionado del microcontrolador PIC18F2685

La Figura 18 muestra el esquemático de conexionado del microcontrolador. En la parte central se encuentra el PIC18F2685 ya mencionado. A su izquierda se encuentra el circuito oscilador hecho por un cristal de cuarzo junto con dos condensadores de 15 pF llevamos a GND (como marca su datasheet [22]) y dos LEDs con su debida resistencia. En la parte inferior se encuentra el conector que se utilizara para comunicarse con la DC2260A explicada anteriormente. Cabe destacar que de este conector hay muchos pines que no se utilizan debido a que corresponden a señales de I2C que no se utilizan. En la parte superior se encuentra el conector necesario para programar el microcontrolador. Finalmente, en la parte derecha está el CAN *transceiver* con su debido conector, el conector para el sensor hall y el conector que deriva la alimentación y tiene dos señales digitales.

#### 4.1.7. Funcionalidades extra

Hasta ahora se ha visto el diseño hardware mínimo, exceptuando el microcontrolador, que tenía que tener el BMS para funcionar correctamente para cumplir los objetivos. Ahora se verán las funcionalidades extras que se han añadido a este diseño.

Primero de todo se han añadido dos LEDs SMD a la placa que deberán ser controlados por el microcontrolador. Puede parecer poco útil un LED, pero a la hora de hacer pruebas de diseño de software son muy útiles y, además, a penas ocupan espacio, por lo que se pueden poner en cualquier

parte. Ésta era la funcionalidad de los LEDs a la hora de diseño de software y testeo de la placa, pero la función final de estos es indicar alertas o errores que se explicaran en el apartado del diseño del software. El único cálculo necesario para la utilización de dichos LEDs es el dimensionamiento de la resistencia necesaria para limitar la corriente. Partiendo de la base que el PIC puede suministrar 20 mA, los necesarios para encender un LED y que la caída de tensión del LED es de 2 V se puede calcular dicha resistencia. Las ecuaciones Ec.8 y Ec.9 muestra el cálculo pertinente de la resistencia necesaria.

$$V_r = V_{microcontrolador} + V_{LED} = 5\text{ V} - 2\text{ V} = 3\text{ V} \quad (\text{Ec. 8})$$

$$R = \frac{V_r}{I_{LED}} = \frac{3\text{ V}}{0,02\text{ A}} = 150\ \Omega \quad (\text{Ec. 9})$$

Dado que el resultado final no es un valor normalizado, se escoge un valor normalizado parecido, en este caso 220  $\Omega$ .

Otra funcionalidad añadida es la de añadir un conector con cuatro salidas: 5V, 0V y dos salidas digitales del microcontrolador. Estas salidas, aunque estén en un mismo conector tiene funcionalidades diferentes. Por una parte, los pines de 5V y 0V, simplemente son una derivación por si en el momento de las pruebas es más cómodo alimentar la placa desde ese punto, por si se quiere medir la alimentación de la placa con un multímetro de una manera sencilla o por si en un futuro dicha alimentación se tiene que derivar a otro módulo BMS. Por otra parte, una de las señales digitales sirve para accionar un conjunto de transistores Darlington para activar o desactivar un relé, de esta manera el BMS podrá decidir si la batería puede cargarse o descargarse en función de su estado. Finalmente, la última señal digital simula la señal BMS OK que dice la normativa de Formula Student.

Por último, se ha añadido un *pin-header* conectado a los puertos RX y TX del microcontrolador que permiten la comunicación con el ordenador mediante el protocolo RS-232. De esta manera se podrá enviar información al ordenador y visualizarla desde allí. Esta función es muy útil a la hora de detectar errores en la programación en la fase de diseño, y puede ser utilizada para enviar los resultados finales a un ordenador.

#### 4.1.8. Diseño del sistema de descarga

La finalidad del sistema de descarga es simular el comportamiento del BMS. Ya que la batería no tiene ninguna aplicación concreta, se simula su descarga con el sistema que se explica a continuación. Además, dicha descarga sirve para comprobar que el BMS mide correctamente los diferentes parámetros de la batería y que es capaz de detectar situaciones de riesgo y actuar en consecuencia.



La descarga de la batería se hace mediante unas resistencias de potencia. Para que no sea una descarga constante sobre una sola resistencia, se han puesto tres en paralelo (Figura 19). De esta manera, además de aumentar la posible corriente total de descarga, se podrá hacer que la batería se descargue a intensidades variables ya que se podrá elegir qué resistencias conectar. También se incorporará un relé que podrá accionar el BMS, de manera que la decisión de descargar más la batería o no, no solo dependerá de un usuario externo sino también del propio BMS. A través de la decisión del BMS sobre el relé se comprobará su funcionamiento.

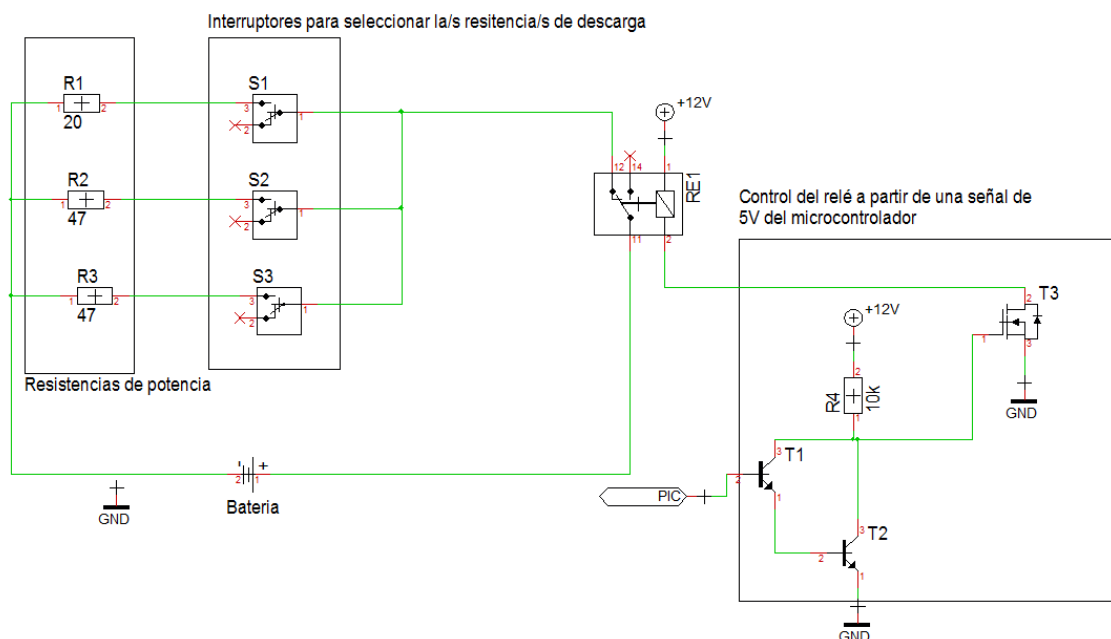


Figura 19 Sistema de descarga de la batería

Las resistencias elegidas son: una de  $20\ \Omega/100\ \text{W}$  y dos de  $47\ \Omega/50\ \text{W}$ . Cuando se conecta la resistencia de  $20\ \Omega$  a la batería totalmente cargada ( $44,4\ \text{V}$ ), pasa una intensidad de  $2,22\ \text{A}$  por dicha resistencia y se disipa una potencia de  $98,57\ \text{W}$ . Cuando se conecta una de las resistencias de  $47\ \Omega$  con la batería cargada, por ésta pasa una intensidad de  $0,94\ \text{A}$  y se disipa una potencia de  $41,94\ \text{W}$ . Debido a que las resistencias pueden llegar a trabajar a una potencia cercana a su límite se recomienda refrigerar mediante un ventilador dichas resistencias cuando se use con la batería cargada para evitar un sobrecalentamiento excesivo. Por tanto, si se conectan las tres a la vez cuando la batería está cargada, se consigue una corriente de descarga de  $4,1\ \text{A}$ .

El relé utilizado es un Gigavac GXNC14CA. Se utiliza ya que era el que proporcionaba e-Tech Racing. Además, dicho relé aguanta unos voltajes e intensidades más que suficientes para este sistema. Cabe destacar que es un relé normalmente cerrado, por lo que se tendrá que tener en cuenta a la hora de programar el software.

Debido a que una señal de 5V que podría proporcionar el microcontrolador no es suficiente para alimentar el relé (que necesita 24V), se ha tenido que pensar una forma de activarlo mediante dicha señal. Por lo que, primeramente, se ha escogido un interruptor (un MOSFET IRF510) que permitiese controlar la alimentación del relé y, posteriormente se ha diseñado la etapa para activarlo. Dicha etapa consiste en una resistencia de *pull-up*, para que, como norma general, el relé este activado y, consecuentemente sus contactos estén abiertos y una estructura Darlington de transistores que cortocircuitaran la resistencia de *pull-up* con GND, cuando reciban una señal del microcontrolador, haciendo que el MOSFET e desactive y corte la alimentación del relé.

## 4.2. Diseño del software

Como se ha dicho anteriormente, el diseño del BMS se divide en dos partes. La parte de hardware ya se ha explicado, por lo tanto, ahora se explicará la programación del microcontrolador. Dicho programa se va a dividir en dos partes: la primera corresponde a la parte necesaria para obtener las medidas de voltaje, corriente y temperatura y actuar en consecuencia y en la segunda se hace la estimación del SoC.

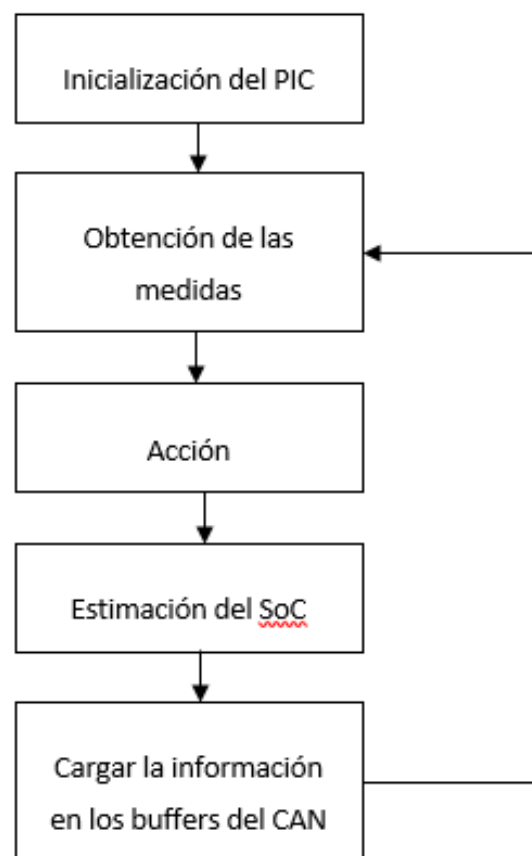


Figura 20 Diagrama general del programa del microcontrolador

El diagrama anterior (Figura 20) muestra de manera muy genérica como se estructura el programa del microcontrolador. Por una parte, se tiene que inicializar, configurando los *fuses*, los *timers*, las interrupciones, las variables, el ADC y las comunicaciones. Después, se tiene que obtener las medidas y actuar en consecuencia ya sea balanceando, encendiendo un LED de advertencia o dando señales digitales. Posteriormente se tiene que estimar el estado de carga de la batería (el SoC). Finalmente, toda la información de las medidas, advertencias, errores y SoC se tiene que cargar en los buffers del CAN para cuando llegue el mensaje de sincronismo, se envíe.

#### 4.2.1. Programa mínimo de obtención de medidas

Este apartado se centra en explicar de forma detallada el programa que se utiliza. Cabe destacar que la estimación del SoC tiene su propio apartado debido a su complejidad, por lo que en este apartado no se tratará.

##### 4.2.1.1. Inicialización

Esta es la primera parte del programa y solo se ejecuta una vez. Primero de todo se indica el microcontrolador que se utiliza, en este caso el PIC18F2685. Posteriormente se configuran los llamados *fuses*, donde se indica que tipo de oscilador se utiliza (cristal de cuarzo), la velocidad de este (20 MHz), si se utiliza circuito de reset o no (no se usa), si se utiliza un *watchdog* o no (no se utiliza) o si se utiliza algún tipo de protección cuando se detecta *undervoltage* (no se utiliza), entre otras cosas. Después, se definen las librerías adicionales que se utilizarán: en este caso, se utilizará la librería de CAN (proporcionada por e-Tech Racing), se asignan nombres a los diferentes pines del microcontrolador para hacer más entendible el programa, se definen las variables que se usarán, se configura el *timer0*, se configuran las comunicaciones SPI (el microcontrolador será la master, se enviarán 8 bits en cada mensaje siendo el primer bit el más significativo, se usarán los pines predeterminados para SPI y la velocidad será de 9600 bps) y el CAN (se hace la inicialización mediante una función de la librería, y se configuran los registros necesarios para configurar la velocidad de esta comunicación en 250 kbps, ya que es la velocidad a la que se comunica toda la electrónica del coche) y, finalmente, se permiten las interrupciones de CAN y del *timer0*.

##### 4.2.1.2. Medidas

Hay tres medidas a hacer: la de voltaje, la de corriente y la de temperatura. El próximo diagrama (Figura 21) mostrará de forma visual y general como se hacen dichas medidas. La primera parte corresponde a la medida de voltaje, la siguiente a la medida de temperatura y la última a la de corriente.

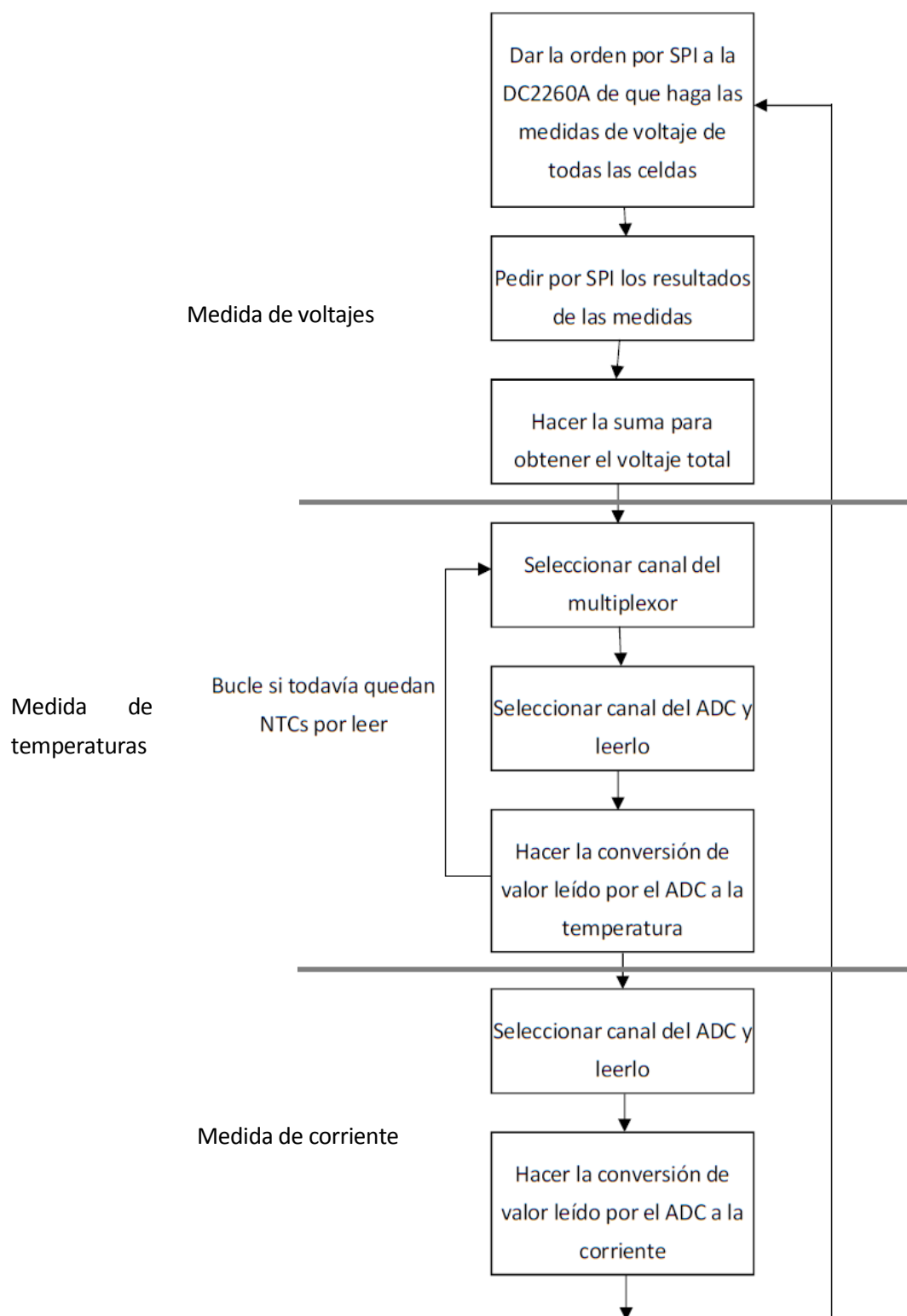


Figura 21 Diagrama de la obtención de medidas

La primera medida que se realiza es la lectura de los voltajes de las celdas. Esta medida la hace el circuito integrado de Linear Technology. Para darle la orden de que haga la medida se le envía vía SPI la orden (que, según el *datasheet*, hay que enviar un 0x83 y un 0x60). Mientras se hacen las lecturas, el PIC está a la espera durante unos pocos milisegundos. Después se le da la orden de leer los resultados, para ello se deben leer 4 registros de 6 bytes del circuito integrado; para leer cada registro se le envía una orden diferente que consta de dos bytes. Cabe destacar que después de cada orden que se le envía a dicho integrado se tiene que enviar dos bytes más que sirven para detectar errores y se calculan con una función proporcionada por el *datasheet* del integrado.

Como se ha explicado anteriormente, las NTCs están multiplexadas para reducir hardware. Por tanto, para leer la temperatura se tiene que hacer un bucle para leer todas las temperaturas. El proceso es el siguiente: se selecciona un canal del multiplexor, se lee el valor del ADC y se hace la conversión. Este proceso se repite hasta haber medido todas las temperaturas. La conversión mencionada consta de transformar el valor leído (un valor que está comprendido entre 0 y 1024) a temperatura. Para ello, primero se calcula el voltaje que ha leído el ADC a partir del valor digital que proporción. Posteriormente con este voltaje, que corresponde al voltaje adaptado para aprovechar todo el rango del ADC mejorando así la resolución, se calcula el voltaje real que hay entre la resistencia serie y la NTC con su resistencia linealizadora en paralelo. Con este valor, se puede obtener el valor de la resistencia de la NTC. Finalmente, sabiendo la resistencia y con la función lineal calculada anteriormente que relaciona la resistencia y la temperatura, se obtiene la medida de la temperatura.

La última medida es la más sencilla. Es la medida de la corriente. Esta medida simplemente se tiene que leer mediante su respectivo canal del ADC del PIC. Una vez que se tiene el valor digital, se calcula el voltaje real que ha medido. Con este voltaje y sabiendo la sensibilidad del sensor hall, se puede calcular la corriente medida teniendo en cuenta que la medida de 0 A corresponde a los 2,5 V leídos por el ADC.

Por último, hay que comentar que al principio y al final del diagrama todavía aparecen unas flechas ya que no es el programa total, sino que corresponde al cuadro de “Obtención de las medidas” del diagrama general del programa. Por tanto, es una parte del programa que proviene de la inicialización y que continua con el cuadro de acciones.

#### 4.2.1.3. Acciones

Este apartado corresponde a las decisiones que tiene que tomar el microcontrolador a partir de las medidas obtenidas anteriormente. El diagrama siguiente (Figura 22) muestra de forma esquemática las acciones que realiza el microcontrolador.

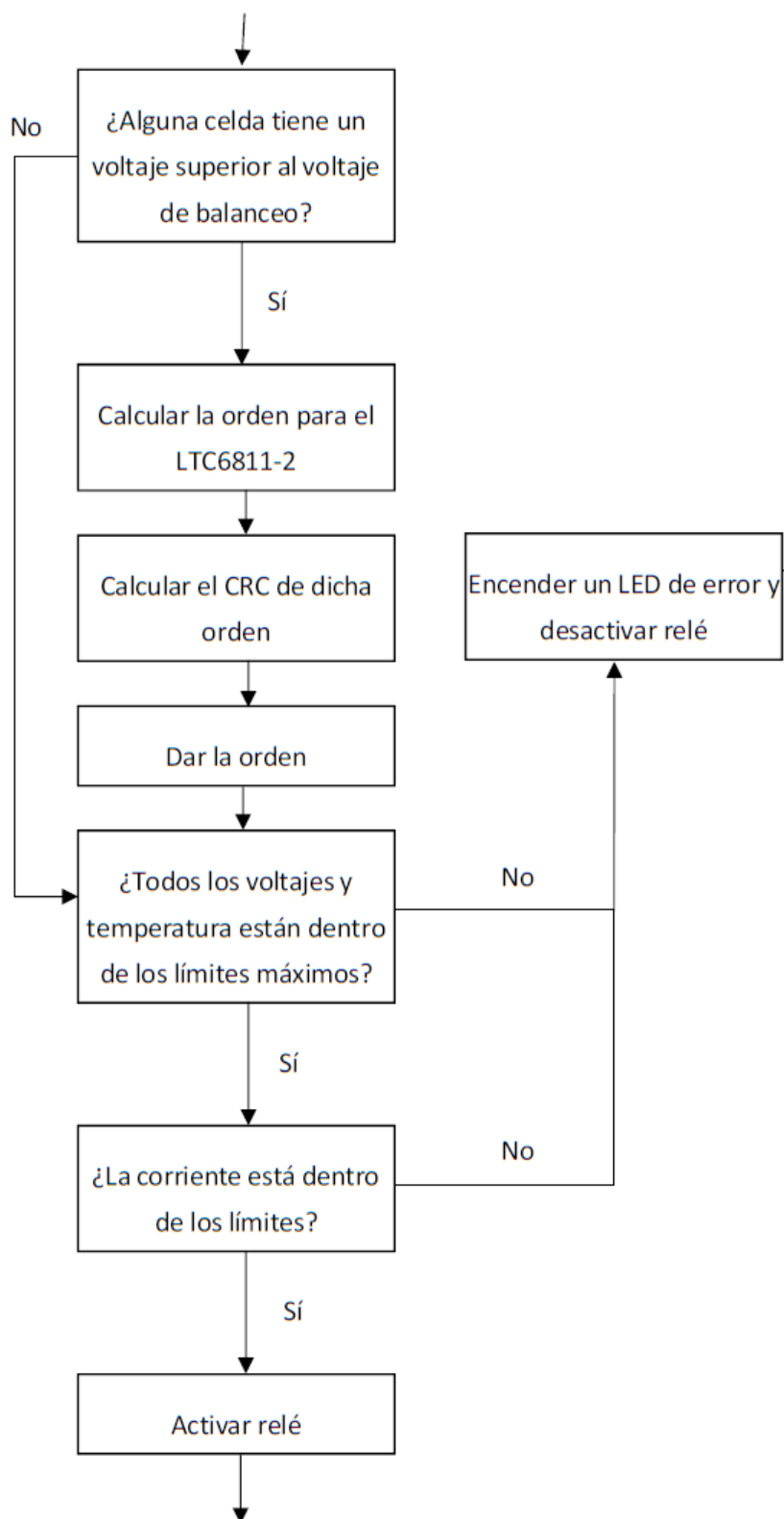


Figura 22 Diagrama de acciones del microcontrolador

Una vez obtenidas todas las medidas, lo primero que analiza el BMS es si tiene que balancear alguna celda o no. Para ello, compara los voltajes de las celdas con un voltaje establecido como límite y si alguna supera dicho límite comienza a balancearse. De esta manera, estableces un voltaje un poco inferior al máximo al cual se considera que la celda está totalmente cargada, y se aseguras que todas las celdas una vez cargadas al máximo lleguen a este voltaje por medio del balanceo. El límite de balanceo deberá ser ligeramente inferior al límite máximo de la celda, para asegurar el correcto funcionamiento del sistema y no perder una carga excesiva. El balanceo se hace mediante el circuito integrado de Linear Technology (LTC6811-2), por lo que se le tiene que dar la orden de que balancee. Para ello se le debe dar la orden mediante unos bytes que dependen de las celdas que haya que balancear, calcular el CRC de dicha orden (que son dos bytes que se envían con la orden que sirven para comprobar que la información que recibe el LTC es correcta) y finalmente enviarle todos estos bytes por SPI.

Una vez hecho esto, se comprueba que todos los valores obtenidos estén dentro de sus límites. Es decir, se verifica que el voltaje o temperatura de la celda no sea inferior o superior a los límites establecidos o que la corriente de la batería sea excesiva ya sea de carga o descarga. Si todos los valores son correctos, se da la señal para activar el relé que permite la descarga por el sistema de descarga explicado anteriormente. Si no están dentro de los límites se enciende un LED y se vuelve al inicio del programa.

Como en el apartado anterior, el primer recuadro y el último tienen flechas ya que este diagrama corresponde al bloque de “acción” del diagrama general del programa. Además, la línea que va de arriba abajo del diagrama corresponde a la línea que va al inicio del programa.

#### 4.2.1.4. Cargar la información en los buffers del CAN

Para no saturar de excesiva información las líneas de CAN del coche, se debe seleccionar muy bien la información que se debe transmitir. Como se ha comentado anteriormente la comunicación CAN se basa en voltajes diferenciales que se interpretan como unos o ceros. Para realizar dicha comunicación, aparte de tener claro la información que se tiene que enviar, se debe definir el identificador y la longitud del mensaje. La información que se ha decidido enviar es el voltaje total de la batería, el voltaje mínimo, el voltaje máximo, la temperatura máxima, la intensidad y el estado de carga. De esta manera la centralita puede controlar el estado de la batería y regular la exigencia de potencia de la batería según el valor de dichos parámetros.

Dado que tanto los voltajes como la intensidad y la temperatura se guardan en variables de 16 bits y que el SoC se almacena en una variable de 32 bits, se tendrán que enviar dos mensajes de CAN ya que solo se pueden enviar 8 bytes de información en cada mensaje. Por esto, cada variable se deberá enviar en partes de 8 bits. Teniendo en cuenta que el identificador debe de ser mayor a 0x80 por regulaciones

del protocolo y debido a que dicho identificador también influye en la importancia y prioridad del mensaje, se ha escogido como identificadores 0x90 y 0x91. En el primer mensaje se enviarán el voltaje máximo, mínimo y total y la intensidad. En el segundo mensaje irá la información correspondiente a la temperatura máxima y el SoC.

Para enviar dicha información, se tienen que definir bien los registros internos del microcontrolador para guardar las variables. De esta manera se debe guardar la información del identificador en los registros *TXBOSIDH* y *TXBOSIDL*, en el primero se guardan los tres bits más significativos del identificador y en el otro el resto. En el registro *TXBODLC* se define la cantidad de bytes a enviar. En el primer mensaje que enviará el BMS este registro se deberá guardar el valor de 8 y en el segundo de 6. Después, en los registros *TXBODX* (siendo X el número de byte a enviar) se guarda la información a enviar. Finalmente, *TXBOCON.txreq = 1*; se pone a uno este bit para que se confirme que el mensaje ha llegado al destino.

#### 4.2.2. Estimación del SOC

Este bloque del diagrama general del programa tiene su propio apartado debido a su complejidad. Además de programar el microcontrolador se tenía que decidir cómo se iba a estimar dicho parámetro. Por tanto, primero, se explicarán las diferentes ideas que se tuvieron y posteriormente cual se eligió y como se programó.

##### 4.2.2.1. Opciones de la estimación del SoC

Se pensó hacer dicha estimación de 4 formas diferentes, 3 de las cuales se basaban en los dos métodos expuestos en el State of the Art, y la otra se basaba en datos experimentales. A continuación, se exponen dichas propuestas con sus ventajas e inconvenientes.

La primera opción consistía en estimarlo mediante test a la celda. Es decir, hacer muchos ensayos de carga y descarga de una celda a diferentes intensidades y temperaturas, de manera que se tuvieran muchos datos de la celda y se pudiese estimar un estado de carga que dependiese de la intensidad y la temperatura. Se debería obtener información sobre la relación entre el voltaje leído y el estado de carga durante cargas y descargas a varias intensidades diferentes y a varias temperaturas, de manera que según la temperatura y la intensidad que leyese el microcontrolador utilizaría una gráfica u otra para estimar el estado de carga. Esta forma de estimarlo sería realmente buena si todas las celdas se comportasen de la misma manera, pero el hecho de que haya pequeñas diferencias celdas hace que no sea muy preciso y, además, no se tienen en cuenta factores como el envejecimiento de las celdas. No se realizó la estimación de esta manera debido a la gran cantidad de ensayos que se le tenían que hacer a la celda, y del tiempo que se disponía.



La segunda opción era combinar el OCV y el Coulomb Counting. De manera que se utilizaría el OCV en la parte inicial y final del voltaje de una batería, donde esta tiene un comportamiento más parecido a una exponencial y el Coulomb Counting en la parte más horizontal. Por un momento se pensó en hacer este método debido a que se tenía conciencia de que había BMS comerciales que lo utilizaban, pero después de analizarlo se decidió descartar. No se utilizó porque es un método que no soluciona ninguno de los grandes problemas de los dos métodos, sino que los magnifica incluso. El OCV no tiene en cuenta la intensidad por lo que, en la parte inicial y final, en el caso de que haya intensidad aparece un error, que se mantendrá y se aumentará con el Coulomb Counting.

El tercer método pensado era utilizar el Coulomb Counting solamente, con la única peculiaridad que la primera vez que se enchufase el BMS como no se sabría el estado actual, se estimase por el OCV. De manera que utilizases el OCV solamente una vez, y a partir de ahí solo se utilizase el Coulomb Counting y se recalibrase cuando la batería está al 100%. El diagrama (Figura 23) siguiente representa dicho algoritmo.

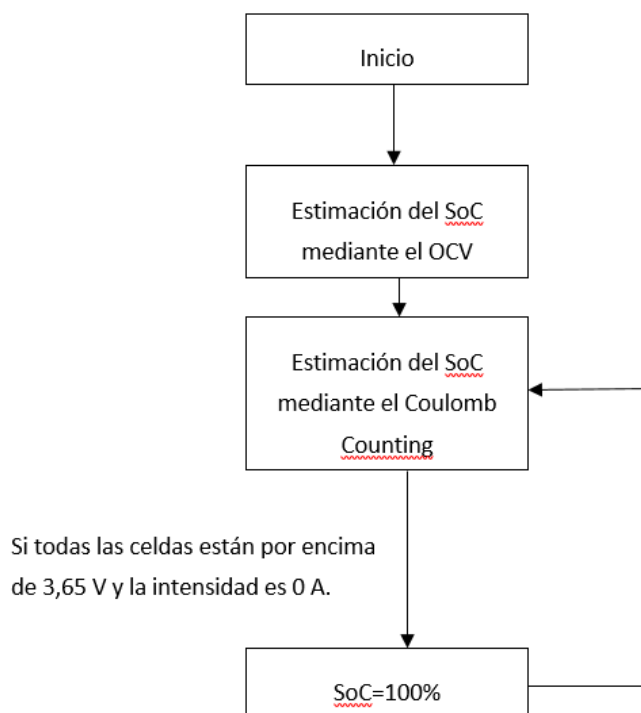


Figura 23 Diagrama de estimación del SoC mediante el Coulomb Counting

Este método se descartó debido a que en ningún momento se tenía en cuenta la temperatura ni la autodescarga de las celdas.

El cuarto método consistía en una combinación absoluta del OCV y el Coulomb Counting teniendo en cuenta la temperatura para el OCV. El siguiente diagrama (Figura 24) muestra cómo se haría la estimación del SoC.

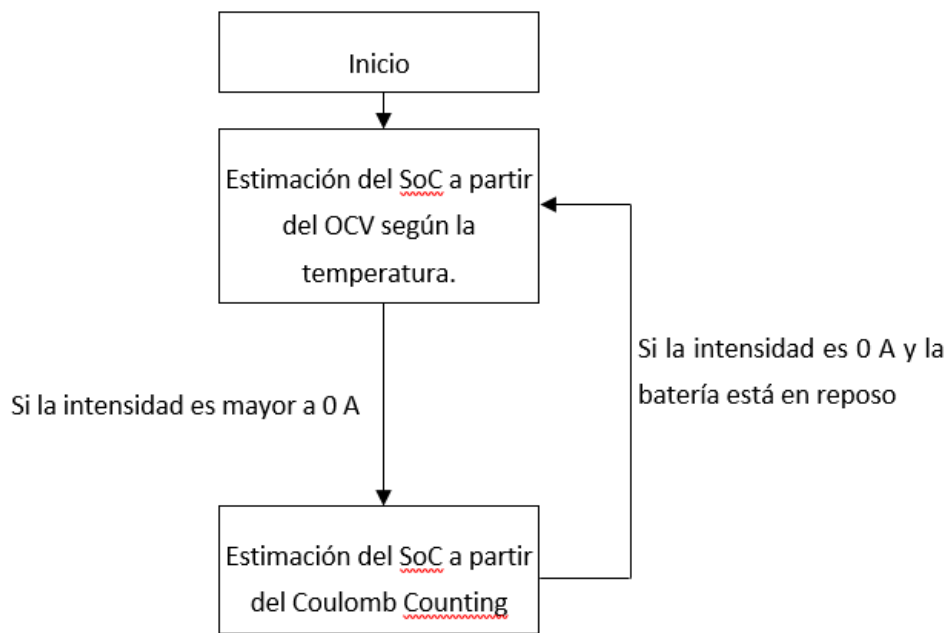


Figura 24 Diagrama de la estimación del SoC seleccionada (combinación del OCV y el Coulomb Counting)

La primera estimación se hace mediante el OCV. La peculiaridad de este OCV es que se calcula la relación del *Open Circuit Voltage* y el SoC a diferentes temperaturas. De esta manera tienes en cuenta la temperatura que ninguno de las otras opciones la tenía en cuenta. Después si el sistema detecta que hay intensidad, el método cambia y pasa a utilizarse el Coulomb Counting. Posteriormente, no hace falta llegar al 100% o al 0% de carga de la batería para recalibrar el Coulomb Counting, sino que se recalibra cuando no se detecta intensidad y, además, ha pasado el tiempo suficiente (en torno a las 4 horas) para considerar que la batería está en reposo.

Cabe destacar que, si este método se quiere utilizar para celdas basadas en otra química, se tendrán que repetir los ensayos de obtención del OCV, ya que, por ejemplo, no se comportan igual unas celdas de LiFePO<sub>4</sub> (como las de este trabajo) que unas LiCO<sub>2</sub> (como las del coche de e-Tech Racing). Por ese motivo, cuando se quiera adaptar este BMS a otro tipo de batería, como sería la del coche de Formula Student de e-Tech Racing, se tendría que repetir todos los ensayos para la obtención del OCV.

#### 4.2.2.2. Obtención del OCV

Como se ha dicho anteriormente, dependiendo del estado de la batería, la estimación del SoC se hace mediante el Coulomb Counting o el OCV. El OCV consiste en relacionar el voltaje en circuito abierto de una celda con su estado de carga. Para obtener dicha relación se han hecho diversos ensayos de carga y descarga a una de las celdas.

Primero, se cogieron las 4 celdas sueltas que se separaron de la batería y se hizo un test de impedancia para comprobar que se podían considerar iguales. Dicho test se ha hecho con el potenciostato Interface 5000E de Gamry y consiste en medir la impedancia de las celdas a diferente frecuencias de manera que al final se puede obtener, entre otros datos, el diagrama de Nyquist de la impedancia de las celdas (Figura 25).

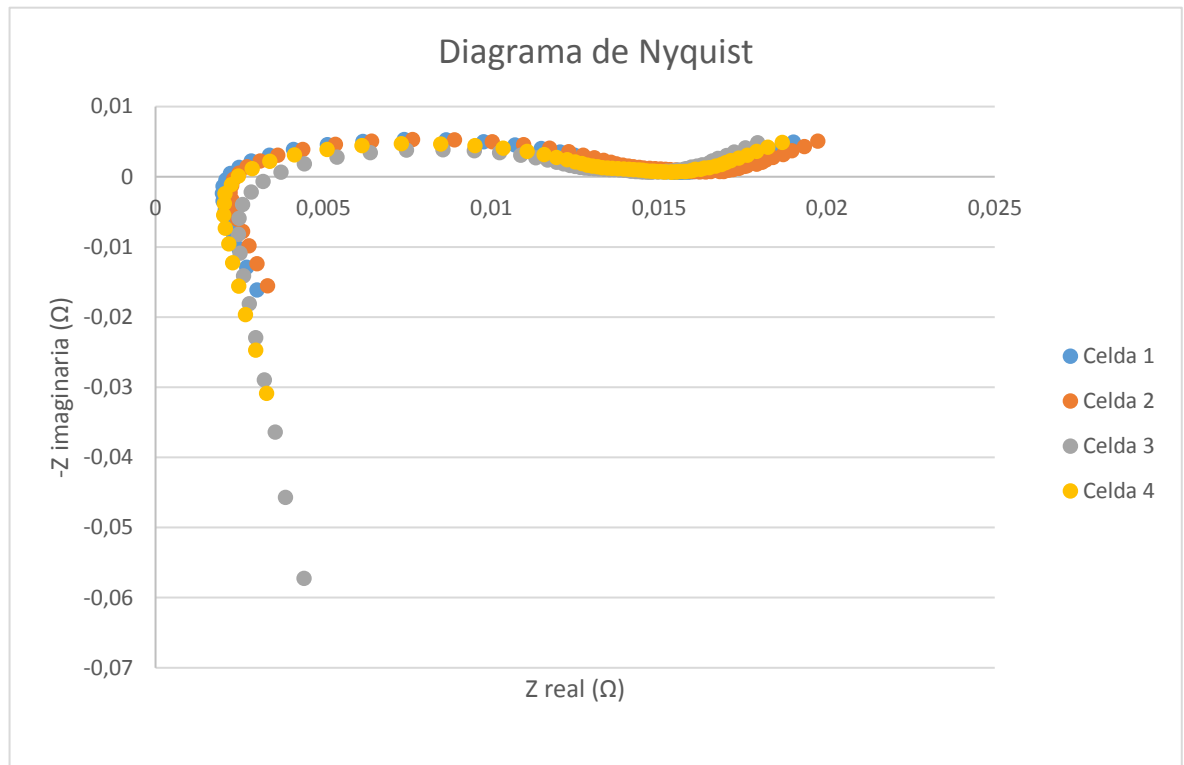


Figura 25 Diagrama de Nyquist de la impedancia de las celdas

Observando la Figura 25 se puede considerar que todas las celdas tienen una impedancia parecida, por lo que se considera que el comportamiento de todas las celdas es igual. Cabe destacar la Z imaginaria positiva (graficada negativamente) corresponde al comportamiento inductivo del conexionado. Después de esta consideración se eligió la celda 1 para hacer los ensayos para obtener el OCV. Estos ensayos se componen de las siguientes fases:

- Medida del voltaje sin cargar ni descargar.
- Carga a corriente constante de 400 mA (C/25).
- Carga a voltaje constante.
- Medida del voltaje sin cargar ni descargar.
- Descarga a corriente constante de 400 mA (C/25).
- Medida del voltaje sin cargar ni descargar durante 4 horas para asegurar que la celda está en reposo.
- Carga a corriente constante de 400 mA (C/25).

- Carga a voltaje constante.
- Medida del voltaje sin cargar ni descargar.

Las fases de medida de voltaje sin carga ni descarga duran 4 horas y sirven para asegurar que la celda está completamente en reposo antes de entrar en la siguiente fase. La primera carga (que se compone por dos fases: la de carga corriente constante y carga a voltaje constante) se utiliza para asegurar que la celda está cargada completamente antes de hacer la descarga. La fase de descarga consiste en descargar a corriente constante la celda desde su máxima carga hasta su mínima. La segunda carga consiste en cargar la celda desde su mínima carga hasta su máxima.

Para la estimación del OCV se utilizan las fases de descarga y la segunda carga. Debido a que se hacen a una intensidad muy baja (400 mA) se puede considerar que el voltaje leído es el mismo que el que tendría la celda en circuito abierto. Como hay pequeñas diferencias entre el voltaje de carga y descarga, se considera el OCV la media entre ambos. Como se ha dicho anteriormente, la capacidad de una celda depende entre otras cosas, de la temperatura. Para tener en cuenta dicho parámetro, estos ensayos presentados se han repetido a diferentes temperaturas, a 20°C, a 40°C y a 60°C. Para asegurar una temperatura constante, la celda permanecía dentro de una cámara térmica (PolEko IL53) durante los ensayos.

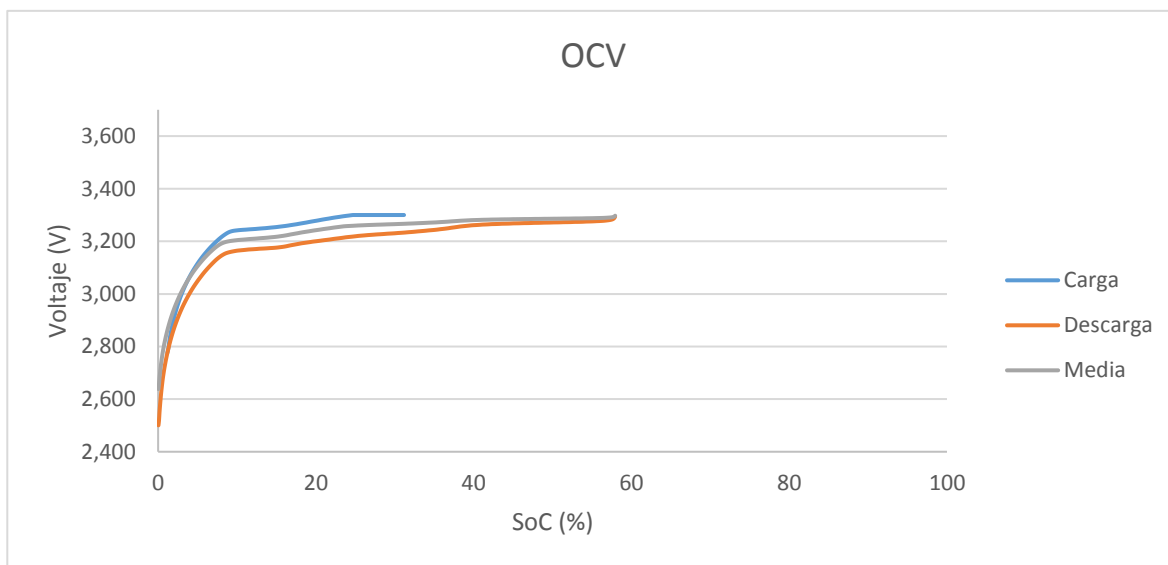


Figura 26 OCV a 20°C

La Figura 26 muestra la relación del SoC y el voltaje de una celda. La gráfica muestra dicha relación tanto en carga como en descarga, pero como se ha dicho anteriormente, la que se utiliza es la media entre las dos debido a que tienen cierta diferencia. Cabe destacar que este ensayo no se realizó correctamente, ya que se puede observar que el SoC no llega al 100%. El motivo de este error fue la mala configuración de los ensayos.

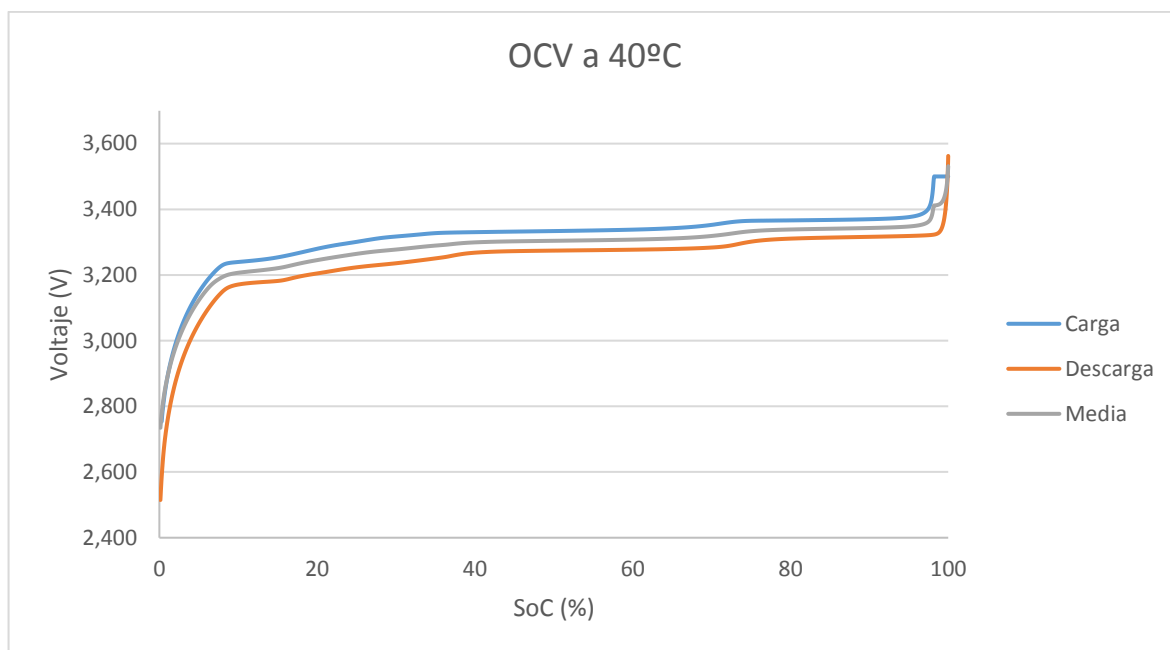


Figura 27 OCV a 40°C

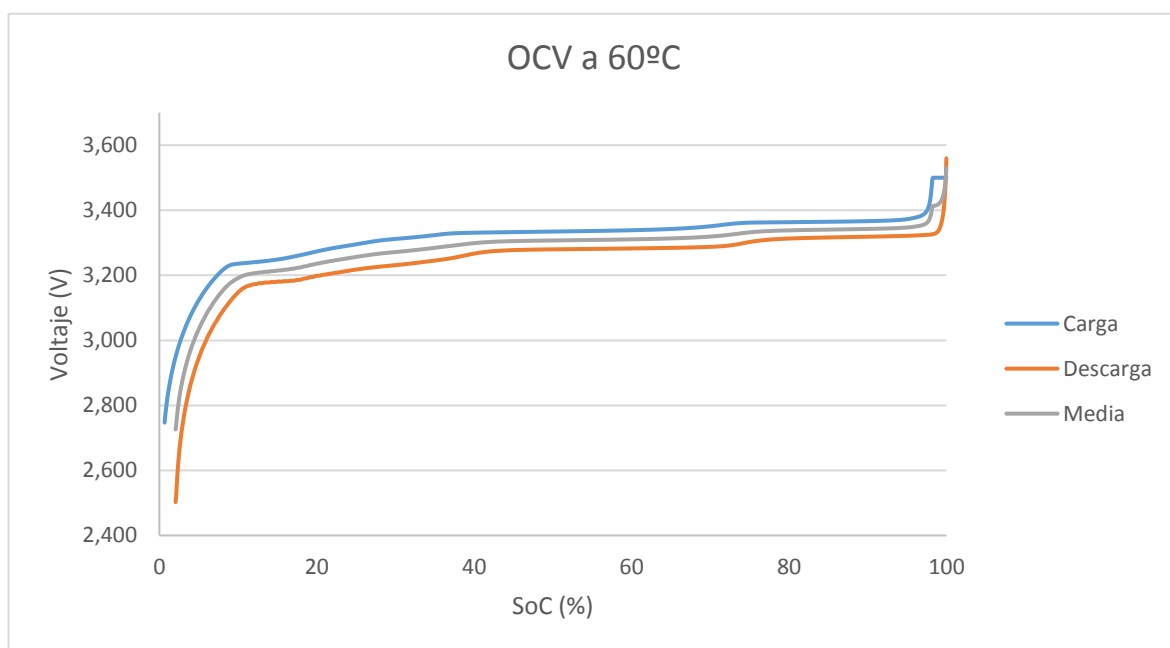


Figura 28 OCV a 60°C

En las Figura 27 y Figura 28 se observa la relación entre el voltaje de la celda y su estado de carga cuando la temperatura era de 40°C y 60°C, respectivamente. Se puede observar en su totalidad el comportamiento del voltaje desde un 0% hasta un 100% en un ciclo completo de vida, entendiendo como ciclo de vida una fase de carga completa y descarga completa.

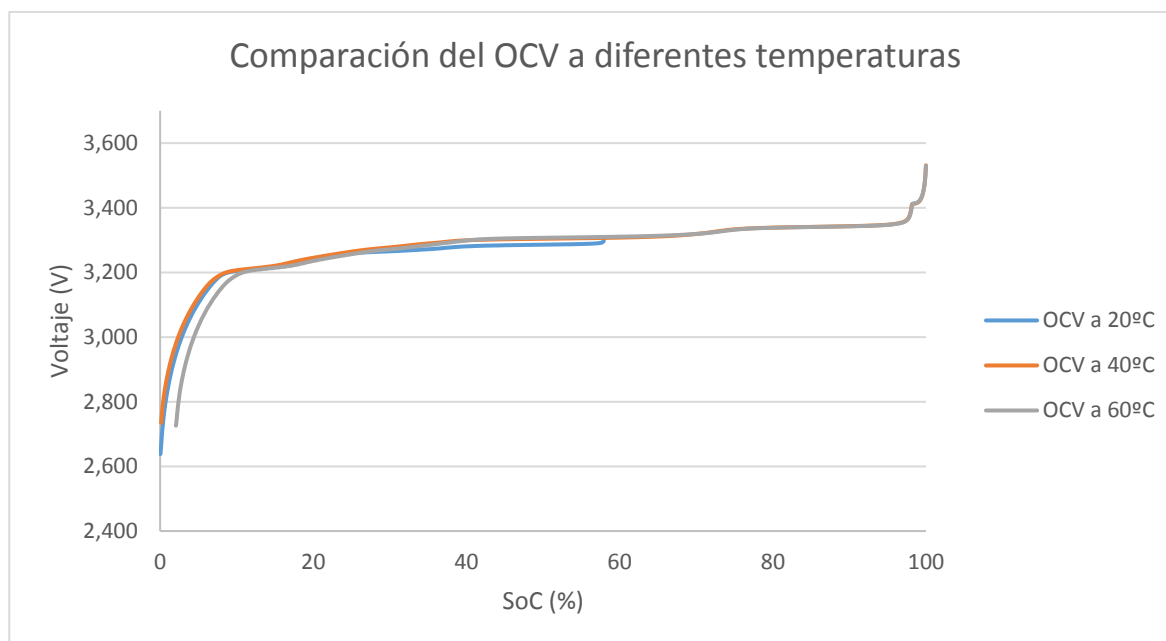


Figura 29 Comparación del OCV a diferentes temperaturas

La Figura 29 muestra en una gráfica el OCV utilizado según la temperatura. Se puede observar que en la parte lineal con el mismo estado de carga la celda que tiene menos voltaje es la que está a 20°C. Por otra parte, la celda que consigue un voltaje mayor es la que estaba a 60°C. Finalmente, se aprecia que en la parte nominal las gráficas de 40°C y 60°C, salvo pequeñas diferencias, prácticamente siguen la misma línea, que la que cae antes de tensión es la celda que estaba a 60°C y que las que estaban en 20°C y 40°C caen más o menos de la misma forma.

#### 4.2.2.3. Programación

Dado que la estimación del SoC se basa en dos métodos se explicará cómo se ha programado cada uno de ellos. Primero, se planteará como se hará la decisión de que método usar. Posteriormente, se explicará la programación del OCV a partir de los datos obtenidos en los ensayos. Finalmente, se explicará la solución adoptada para aplicar el Coulomb Counting.

Como mostraba el diagrama de estimación del SoC adoptado, se utilizarán dos métodos dependiendo de la corriente y del tiempo de reposo de la batería. Por una parte, si se detecta corriente, el programa directamente utilizará el método del Coulomb Counting y pondrá un contador a 0. Por otra parte, si el programa no detecta corriente y este contador, que se incrementa a través de un timer para asegurar el tiempo de reposo de la batería, ha llegado al valor correspondiente a las 4 horas o es la primera vez que se ejecuta se utilizará el OCV.

Las gráficas anteriores del OCV se han realizado a partir de los resultados de los ensayos, los cuales venían dados en tablas de datos. Dichos datos se han utilizado para obtener el voltaje medio entre la

carga y la descarga. Posteriormente, esos datos de voltajes medios y su SoC correspondiente se han introducido en el programa del microcontrolador como tablas de constantes, es decir, una tabla de valores que no se puede modificar (*look-up table*). La batería está compuesta de 12 celdas en serie, por lo que se entiende que su estado de carga viene determinado por el estado de carga más bajo de sus celdas. Por ese motivo, primero se determina cuál es la celda con menos voltaje. Después, a partir de la temperatura leída, se decide si hacer la relación entre el voltaje y el SoC con una tabla u otra. Si la temperatura es de 25°C se estimará el SoC a partir de la tabla de 20°C, por el contrario, si la temperatura es de 55°C se estimará el SoC a partir de la tabla de 60°C.

Para hacer la estimación del Coulomb Counting lo que se hace es que cada 100 ms se mira la intensidad medida y se calcula la cantidad de carga que ha perdido la batería en ese pequeño periodo de tiempo con esa intensidad. Sabiendo la carga perdida y la capacidad total de la batería se puede saber en porcentaje la carga perdida.





## 5. Fabricación

Esta apartado abarca todo proceso y métodos que se han hecho para realizar el proyecto. Primero se explica la adaptación de la batería y después se explicará la fabricación de la PCB y del sistema de descarga. Durante dichas explicaciones también se tendrá en cuenta el cableado necesario para hacer todo el sistema funcionar.

La batería tenía su propio encapsulado tal como muestra la Figura 30 **Error! No se encuentra el origen de la referencia..** Esto no interesaba ya que dentro estaba la batería compuesta por 16 celdas y su propio BMS. Por tanto, se quitó la batería de esta caja y se desenchufó el conector del BMS. En ese momento, se tenían 16 celdas LiFePO4 conectadas en serie. Las celdas estaban unidas mediante pequeñas pletinas atornilladas, por lo que se desatornillaron las 4 que se iban a utilizar para testearlas y, finalmente, quedo lo que en un principio se quería: una batería compuesta por 12 celdas totalmente en serie (configuración 12s1p).



Figura 30 Batería LFP-4810S 48V/10AH

El conector de dicha batería a su BMS no era compatible con el que hay en la DC2260A. Por ese motivo se cortaron los cables de dicho conector, se alargaron con manguitos preaislados de Sofamel, y finalmente se atornillaron en el conector Weidmüller correspondiente. Por otra parte, los terminales positivos y negativos de la batería se soldaron a un conector aéreo de manera que se pudiese enchufar de manera sencilla al circuito de descarga o a una fuente externa para cargar la batería. La Figura 31 muestra el resultado final de la conexión entre la batería y la demoboard.

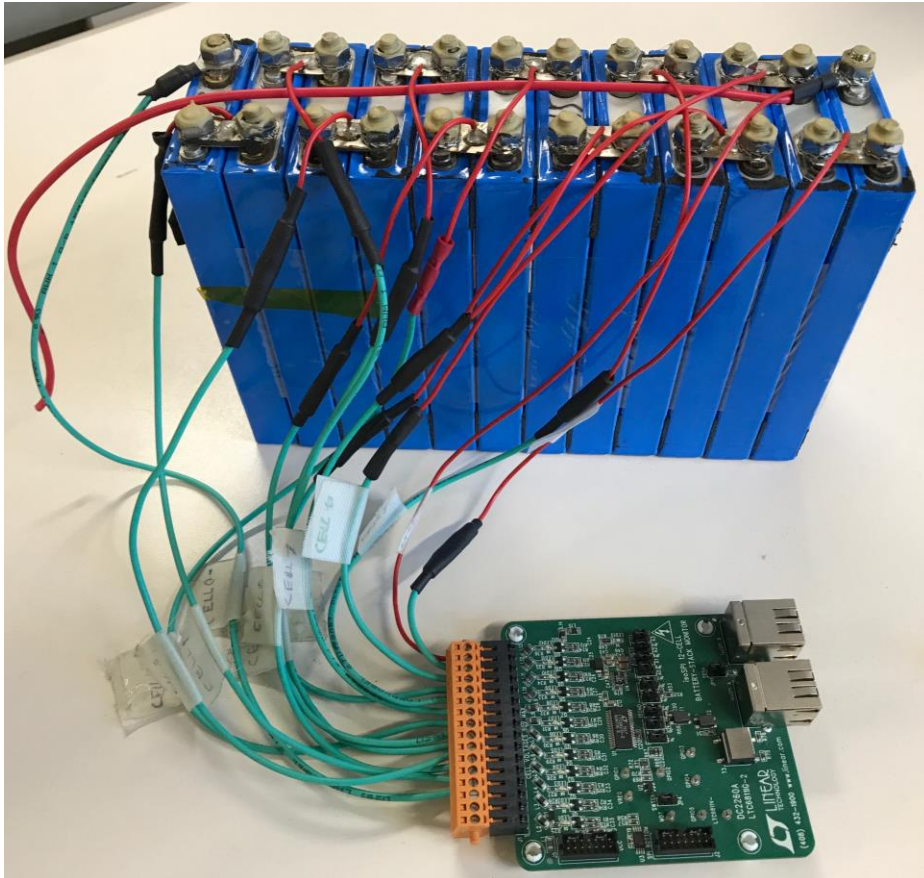


Figura 31 Bateria y demoboard de Linear Technology cableados

Todo el diseño explicado sobre el hardware del BMS se debía construir de alguna manera. La forma de hacerlo era mediante la fabricación de una PCB (*Printed Circuit Board*) (Figura 32). Se pensó hacerlo de manera “casera”, mediante la impresión del fotolito y los procesos químicos necesarios. Debido a la poca experiencia y a la no disponibilidad de todos los elementos necesarios para hacer dicha PCB, se pidió a unos profesores de la universidad que la hicieran. El hecho de hacerlo “casero” tiene bastantes inconvenientes: las pistas pueden no salir todo lo bien que se quería, hay muchas limitaciones de diseño adicionales, la placa no tendría protección ninguna (el cobre queda directamente al aire) y las vías se tienen que taladrar a posteriori poniendo en riesgo las propias pistas si no se hace bien. Debido a estos inconvenientes y al hecho de que hubo un fallo de diseño a la hora de hacerla, se corrigió la placa y se mandó a hacer a una empresa china, PCBCart. Una vez se dispuso de la placa, se soldaron todos los componentes y se empezó la programación. Cabe destacar que, aunque ya se hubiesen probado muchas de las funcionalidades del sistema antes de mandarla a hacer, se tenía que pulir muchos detalles y comprobar que la placa en si no daba ningún problema.

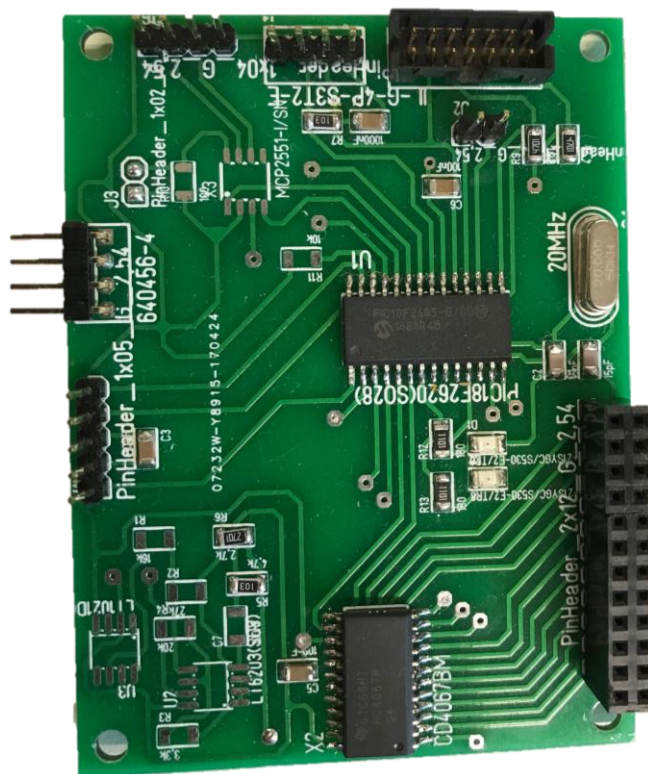


Figura 32 PCB final

Finalmente, tanto la batería con la demoboard y la PCB se tenía que conectar. Además, se tenía que cablear las NTCs, soldar el sensor hall a los cables de la batería y conectarlo a la PCB. La muestra el resultado final del montaje del BMS definitivo.

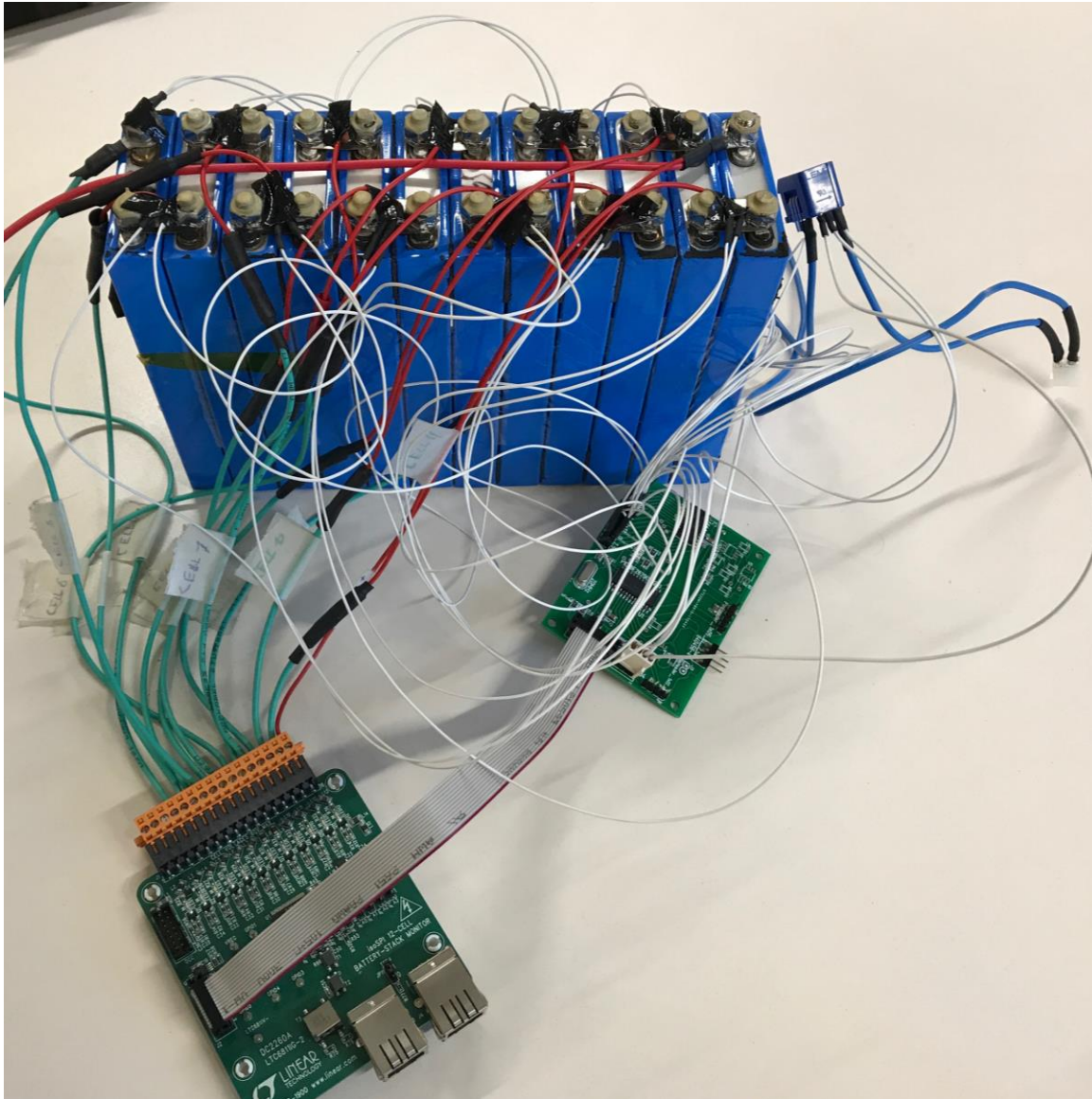


Figura 33 Montaje final del BMS

El sistema de descarga, que se usó para probar el correcto funcionamiento del BMS se montó en una plancha de madera. En dicha madera se atornillaron las tres resistencias de potencia seleccionadas y el relé. Posteriormente se hicieron las conexiones necesarias entre las resistencias, el relé, los interruptores y el conector para la batería. Finalmente, se soldó el control del relé en una placa de topos, ya que, en el momento de diseñar la placa no se había pensado en él, y se soldó el sensor hall a los cables correspondientes.



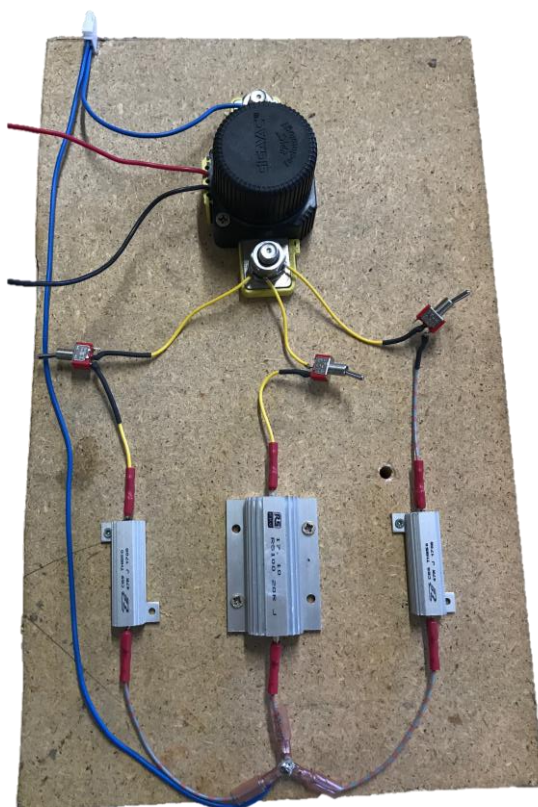


Figura 34 Sistema para descargar la batería

Una vez se tenían todos los elementos, era el momento de interconectarlos e alimentarlos. La Figura 35 muestra el resultado final del montaje en pleno funcionamiento.

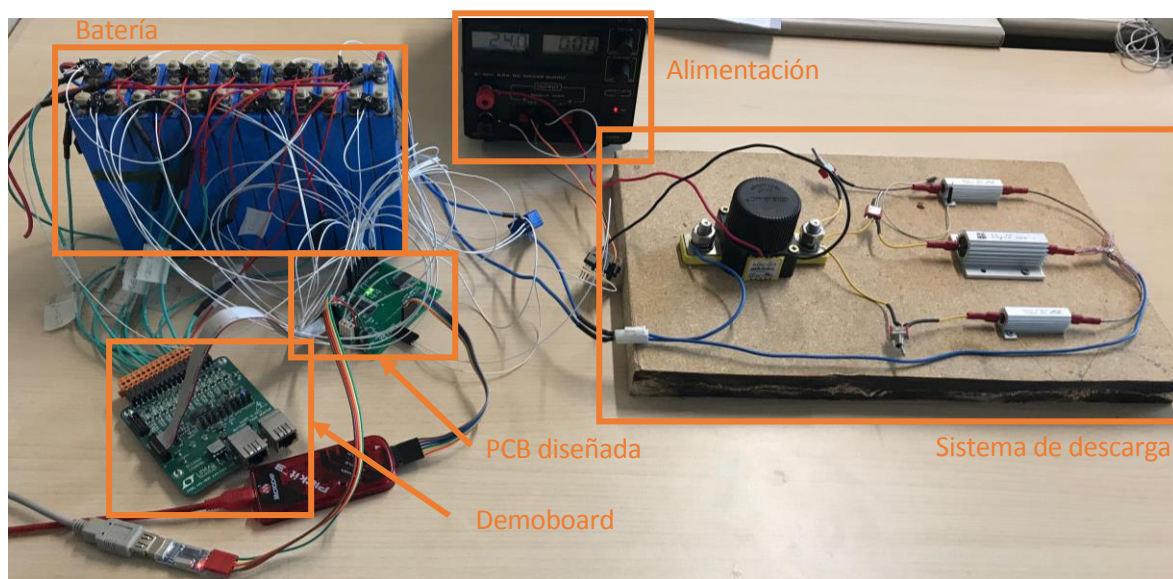


Figura 35 Montaje final de la batería con el BMS y el sistema de descarga



## 6. Resultados

En este apartado, se presentan los resultados obtenidos con el diseño del BMS. Por una parte, se obtienen los resultados de las medidas correspondientes de voltaje, corriente y temperatura y se comparan con mediciones de otros aparatos. Por otra parte, se prueba la estimación del SoC mediante la descarga completa de la batería y se analizan los datos obtenidos. Para visualizar los datos obtenidos por el BMS, se ha usado la comunicación RS-232 del microcontrolador para enviar información al ordenador y visualizarlo mediante el programa Hercules.

### 6.1. Resultados de la prueba de medición de voltajes

Esta prueba se ha realizado, conectando la batería a la demoboard de Linear Technology y ésta a la PCB diseñada. No se han tenido en cuenta ni las NTCs ni el sensor hall de corriente.

```
El voltaje de 1 es 3274 mV
El voltaje de 2 es 3272 mV
El voltaje de 3 es 3277 mV
El voltaje de 4 es 3269 mV
El voltaje de 5 es 3267 mV
El voltaje de 6 es 3272 mV
El voltaje de 7 es 3270 mV
El voltaje de 8 es 3269 mV
El voltaje de 9 es 3274 mV
El voltaje de 10 es 3271 mV
El voltaje de 11 es 3266 mV
El voltaje de 12 es 3270 mV
El voltage total es 39251
```

Figura 36 Captura de pantalla del Hercules que muestran los voltajes medidos

La Figura 36 muestra las lecturas de voltaje obtenidas por el BMS. Para comparar los resultados se ha usado un multímetro de HT Instruments, el HT64. Dicho multímetro a medido los siguientes voltajes:

- Voltaje de la celda 1: 3273 mV
- Voltaje de la celda 2: 3271 mV
- Voltaje de la celda 3: 3275 mV
- Voltaje de la celda 4: 3267 mV
- Voltaje de la celda 5: 3265 mV
- Voltaje de la celda 6: 3272 mV
- Voltaje de la celda 7: 3269 mV
- Voltaje de la celda 8: 3269 mV
- Voltaje de la celda 9: 3273 mV
- Voltaje de la celda 10: 3269 mV
- Voltaje de la celda 11: 3265 mV
- Voltaje de la celda 12: 3270 mV

- Voltaje total de la batería: 39240 mV

Como se puede observar el error máximo de voltaje de una celda es de 2 mV, lo que ha supuesto un error en la medida del voltaje total de 11 mV. Vistos estos resultados que se puede considerar una medida más que fiable la que hace el LTC6811 de Linear Technology.

## 6.2. Resultados de las mediciones de temperatura

Para comprobar el correcto funcionamiento de estas medidas solo se ha conectado una NTC a la PCB para simplificar tanto cableado como la programación. De esta manera, las posibles correcciones de software se hacían mucho más ágiles. En las Figura 37 y Figura 38 se muestran una sencillas pruebas que se hicieron para comprobar el correcto funcionamiento de la medición de la temperatura.

```
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 27.21 °C
-----La temperatura leida es= 26.62 °C
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 24.85 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 24.85 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 26.03 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 25.44 °C
-----La temperatura leida es= 26.62 °C
-----La temperatura leida es= 25.44 °C
```

Figura 37 Captura de pantalla del Hercules que muestran la temperatura medida por una NTC a temperatura ambiente

Como se observa en la Figura 37 la temperatura medida oscila entre 25 °C y 26 °C, más o menos. Esta prueba se hizo a temperatura ambiente y para comparar los resultados se midió dicha temperatura con un termopar K conectado a una pinza amperimétrica/multímetro, la HT4013 y con una pistola de infrarrojos de PCE Instruments, la PCE-777. La primera medía una temperatura de 23,3-23,6 °C y la segunda media 26 °C apuntando a la NTC. La posible tolerancia de la NTC y de las demás resistencias y la aproximación que se asume con la linealización hacen que el error total de la medida de temperatura a temperatura ambiente sea considerablemente alto.



```

-----La temperatura leída es= 30.14 °C
-----La temperatura leída es= 31.30 °C
-----La temperatura leída es= 31.88 °C
-----La temperatura leída es= 34.18 °C
-----La temperatura leída es= 33.03 °C
-----La temperatura leída es= 33.03 °C
-----La temperatura leída es= 33.03 °C
-----La temperatura leída es= 33.03 °C
-----La temperatura leída es= 32.45 °C
-----La temperatura leída es= 33.60 °C
-----La temperatura leída es= 33.03 °C
-----La temperatura leída es= 33.03 °C
-----La temperatura leída es= 33.03 °C
-----La temperatura leída es= 32.45 °C
-----La temperatura leída es= 32.45 °C
-----La temperatura leída es= 33.03 °C
-----La temperatura leída es= 33.60 °C
-----La temperatura leída es= 32.45 °C
-----La temperatura leída es= 32.45 °C
-----La temperatura leída es= 32.45 °C
-----La temperatura leída es= 32.45 °C
-----La temperatura leída es= 32.45 °C
-----La temperatura leída es= 35.32 °C
-----La temperatura leída es= 33.03 °C
-----La temperatura leída es= 32.45 °C
-----La temperatura leída es= 33.03 °C
-----

```

Figura 38 Captura de pantalla del Hercules que muestran la temperatura medida por una NTC apretándola con los dedos

En la Figura 38 se muestra que la temperatura medida por la NTC mientras ésta está siendo apretada por los dedos de una persona es aproximadamente 32°C. Para comparar dicha temperatura, en este caso solo se pudo utilizar el termopar por razones lógicas. La temperatura medida por dicho termopar fue de 30,8°C. Se observa que a una temperatura superior a la temperatura ambiente la precisión de la NTC aumenta y reduce el error.

### 6.3. Pruebas mediciones de corriente

Para comprobar la medición de corriente lo que se hizo fue conectar una resistencia de potencia de 20  $\Omega$ /100 W a una fuente de tensión de voltaje regulable (modelo y marca), de manera que según el voltaje que proporcionaba daba una intensidad u otra. La Figura 39 se puede observar los datos obtenidos de la prueba de medición de corriente.

```

-----La corriente leída es= -0.01 A
-----La corriente leída es= -0.19 A
-----La corriente leída es= -0.18 A
-----La corriente leída es= -0.16 A
-----La corriente leída es= -0.07 A
-----La corriente leída es= -0.12 A
-----La corriente leída es= 0.04 A
-----La corriente leída es= -0.16 A
-----La corriente leída es= -0.09 A
-----La corriente leída es= -0.15 A
-----La corriente leída es= -0.61 A
-----La corriente leída es= -0.85 A
-----La corriente leída es= -0.61 A
-----La corriente leída es= -0.86 A
-----La corriente leída es= -0.79 A
-----La corriente leída es= -0.72 A
-----La corriente leída es= -0.72 A
-----La corriente leída es= -1.30 A
-----La corriente leída es= -1.47 A
-----La corriente leída es= -1.27 A
-----La corriente leída es= -1.21 A
-----La corriente leída es= -1.28 A
-----La corriente leída es= -1.33 A
-----La corriente leída es= -1.39 A
-----La corriente leída es= -1.36 A
-----La corriente leída es= -1.19 A
-----La corriente leída es= -1.33 A
-----La corriente leída es= -1.30 A
-----

```

Figura 39 Captura de pantalla del Hercules de la prueba de medición de corriente

Para entender los datos obtenidos hay que destacar que no corresponden a una intensidad constante, sino que hay tres franjas diferenciadas la primera corresponde al reposo (0A), la segunda corresponde a una intensidad media y la última corresponde a la máxima intensidad que podía aportar la fuente de tensión que se usaba para alimentar la resistencia. Cabe destacar que el signo negativo de la intensidad viene dado por la orientación del sensor. En la primera franja se observa que la corriente leída tiene una variación máxima de 200 mA. Por otra parte, en la siguiente franja se lee una intensidad que oscila entre 0,6 A y 0,8 A aproximadamente. En esta franja la fuente de tensión marcaba una intensidad de 0,6 A y una pinza amperimétrica (la ya usada para medir la temperatura, la HT4013) medía 0,67 A. Por último, en la última franja, el microcontrolador mediante el sensor hall media 1,39 A, más o menos, mientras que la fuente marcaba 1,15 A y 1,24 A por la pinza amperimétrica. Como se puede observar, ninguno de los tres resultados coincide, pero se debe tener en cuenta que el más dispar es el medido por el BMS. Puede haber varios motivos para dicho posible error de la medida. El primero podría ser debido a pequeñas interferencias electromagnéticas, ya que pocos milivoltios de interferencias provocan lecturas de varias decenas de miliamperios. El segundo es que la alimentación del hall no fuera del todo correcta y entonces la referencia de este pudiera estar descentrada, lo que explicaría que el error siempre fuese de aproximadamente 0,2 A. El tercero y último podría ser una combinación de los dos anteriores.

Como conclusiones de la prueba de temperatura y corriente se puede decir que el error de una medición, no solo viene de los cálculos teóricos, sino que también viene de la tolerancia de los

elementos que se utilizan, del correcto cableado y las buenas conexiones y de muchos más factores que no se tuvieron del todo en cuenta en este diseño. Estos aspectos se deberán tener en cuenta para futuros estudios.

## 6.4. Pruebas finales de las mediciones

Finalmente, se unificaron todos los programas de medición y se optimizaron un poco viendo los resultados obtenidos. En la medida de la temperatura se introdujo un filtro pasa-bajos realizado mediante software para darle cierta estabilidad a la medida. En la medida de corriente, se intentó introducir el mismo filtro pasa-bajos, pero no dio los resultados esperados. Por lo que se recalibró a partir de corrientes conocidas y sabiendo se aproximó la relación entre el valor que leía el microcontrolador y el valor de corriente. Además, a la corriente se le añadió una condición que consistía en que toda lectura que superara los -6,5 A o 6,5 A será descartada ya que el sistema nunca trabajará a intensidades superiores a estas, por lo que dichas medidas vendrían provocadas por posibles interferencias o malas lecturas

```
El voltaje total es: 38650 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 64 mA
-----
El voltaje total es: 38652 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 319 mA
-----
El voltaje total es: 38655 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 86 mA
-----
El voltaje total es: 38659 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 77 mA
-----
El voltaje total es: 38661 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 86 mA
-----
```

Figura 40 Captura de pantalla del Hercules que muestran las mediciones generales de la batería

En la Figura 40 se muestran las medidas obtenidas con la batería en reposo y en temperatura ambiente. El voltaje leído por el multímetro iba subiendo (marcando aproximadamente los mismos valores) como la lectura del microcontrolador debido a que se había estado descargando anteriormente y por eso ahora estaba recuperando. La temperatura que muestra corresponde a 24,15 °C y con la sonda K se midió una temperatura de 22,93 °C. Esta medida supone cierta mejora respecto a las pruebas anteriores, ya que el hecho de aplicar el filtro pasa-bajos y el de hacer la media de 12 mediciones ha mostrado que el error total fuese inferior a los 2°C que se tenía anteriormente a temperatura ambiente. Por último, la medida de corriente, cuando teóricamente debería marcar 0 A, tiene ciertas oscilaciones, pero el error se ha reducido hasta menos de 100 mA. Visto que las medidas anteriores

tenían un error aproximadamente de 200 mA, se considera que estas últimas medidas son válidas para el BMS.

```

El voltaje total es: 38359 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 819 mA
-----
El voltaje total es: 38346 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 803 mA
-----
El voltaje total es: 38334 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 773 mA
-----
El voltaje total es: 38323 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 725 mA
-----
El voltaje total es: 38314 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 803 mA
-----
El voltaje total es: 38310 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 959 mA
-----
El voltaje total es: 38302 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 789 mA
-----

```

*Figura 41 Captura de pantalla del Hercules que muestran las mediciones generales de la batería descargando a 815mA*

En la Figura 41 muestra las mediciones generales de la batería mientras está se descargaba a una intensidad teórica de 815 mA según el voltaje de la batería (38,3 V) y la resistencia (47  $\Omega$ ). La pinza amperimétrica de HT marcaba 0,78 A y la medida del microcontrolador oscila 100 mA para arriba y menos de 50 mA para abajo. Se puede considerar que esta medida de corriente es apta para la aplicación del BMS.

## 6.5. Pruebas de balanceo

Para comprobar que el BMS era capaz de balancear pasivamente las celdas se modificaron los límites de voltajes de manera que en la situación de la batería entrase en modo balanceo. Los voltajes de las celdas variaban entre 3,272 V y 3,281 V. Primero se estableció que toda celda con un voltaje superiora 3,2 V, balanceara.

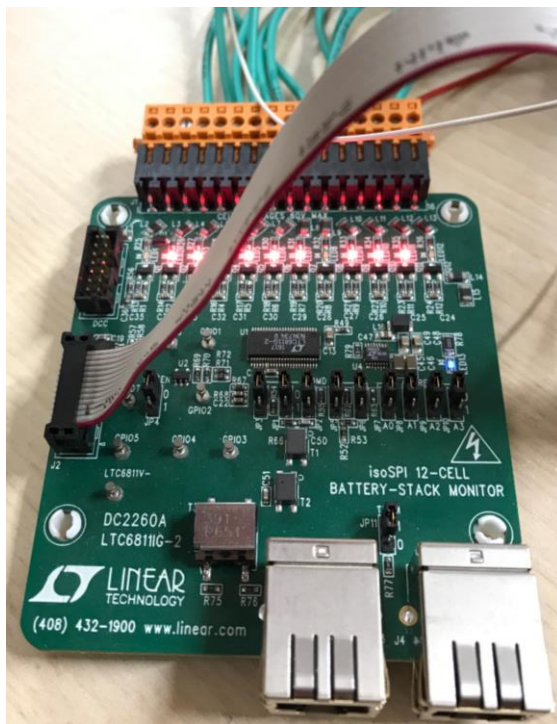


Figura 42 Comprobación del correcto balanceo de todas las celdas

La Figura 42 muestra el resultado obtenido con la configuración anterior. Se puede observar mediante la iluminación de los LEDs que todas las celdas empiezan a balancear. Posteriormente, para garantizar el correcto funcionamiento del balanceo, se estableció un voltaje de 3,279. La Figura 43 muestra el resultado de dicha prueba. Se puede observar que solo dos de las doce celdas comienzan a balancear. El voltaje de éstas era 3,281 V y 3,280 V, por lo que se puede afirmar que el sistema de balanceo funciona correctamente.

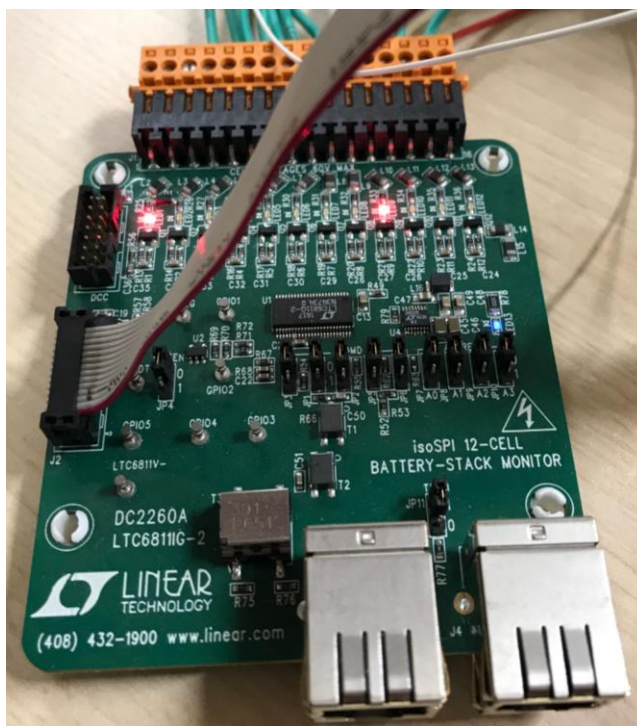


Figura 43 Comprobación de balanceo de celdas que estaban por debajo de 3,279 V

## 6.6. Pruebas de la estimación del SoC

Este test muestra los resultados obtenidos a partir de la forma de estimar el SoC que se ha decidido utilizar. Cabe destacar que no se tenía ninguna medida con la que comparar, así que la decisión final de si es correcta o no de dicha forma de estimar el SoC, se obtendrá una vez se hayan completado varios ciclos de la batería con el BMS conectado.

### 6.6.1. Estimación del SoC a partir del OCV

Primero, se ha visualizado la estimación del SoC cuando la batería está en reposo. Para ello, como se ha definido antes, simplemente, según la temperatura, se ha relacionado un valor del SoC con el voltaje mínimo de celda mediante *look-up tables*.

```

El voltaje total es: 39239 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 0 mA
El estado de carga de la batería es de: 4183 (unidades en tanto por diez mil)
-----
El voltaje total es: 39240 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 8 mA
El estado de carga de la batería es de: 4183 (unidades en tanto por diez mil)
-----
El voltaje total es: 39239 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 11 mA
El estado de carga de la batería es de: 4183 (unidades en tanto por diez mil)
-----
El voltaje total es: 39239 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 11 mA
El estado de carga de la batería es de: 4183 (unidades en tanto por diez mil)
-----
El voltaje total es: 39240 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 2 mA
El estado de carga de la batería es de: 4183 (unidades en tanto por diez mil)

```

Figura 44 Captura de pantalla del programa Hercules que muestra los datos más relevantes de la batería

Como se puede observar en la Figura 44 el SoC estimado de la batería es de 41,83%. Observando las tablas a partir de las cuales se han realizado las gráficas del OCV, se puede comprobar que dicha estimación es correcta. En la Figura 45 se muestra la relación entre el SoC y el voltaje a 20°C de manera acotada para poder apreciar dicha relación. El voltaje mínimo leído con el multímetro de HT es de 3,264 y, según las tablas, con 3,265(voltaje leído por el microcontrolador) le corresponde el estado de carga de 41,83% (mínimo de los posibles). Cabe destacar que el BMS trabaja con decimales para mejorar la exactitud de la estimación, pero al final la unidad que verá el piloto será en múltiplos de 1%.

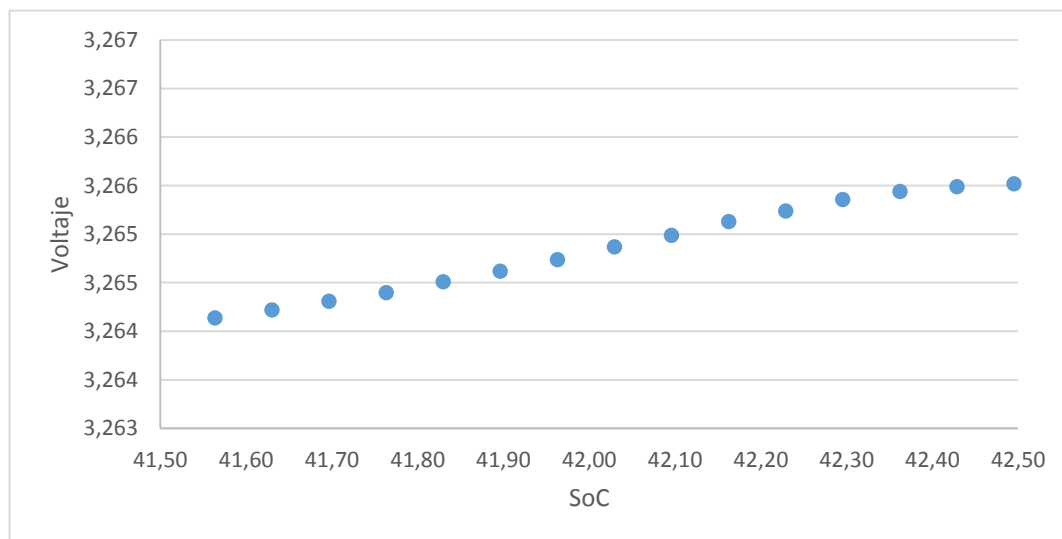


Figura 45 Gráfica acotada de la relación del voltaje con el SoC a 20°C



### 6.6.2. Estimación del SoC a partir del Coulomb Counting

Para hacer esta prueba se ha conectado la batería al sistema de descarga, de manera que se fuese descargando. En la Figura 46 muestra de forma simbólica parte del resultado de esta prueba. Hay seis datos que se han obtenido cada 15 segundos a una intensidad de 1,82 A según la pinza amperométrica de HT.

```

El voltaje total es: 37605 mV
La temperatura media es: 2480 (unidades en 0,01°C)
La intensidad es: 1873 mA
El estado de carga de la batería es de: 2144 (unidades en tanto por diez mil)
-----
El voltaje total es: 37587 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 1735 mA
El estado de carga de la batería es de: 2140 (unidades en tanto por diez mil)
-----
El voltaje total es: 37577 mV
La temperatura media es: 2480 (unidades en 0,01°C)
La intensidad es: 1804 mA
El estado de carga de la batería es de: 2136 (unidades en tanto por diez mil)
-----
El voltaje total es: 37568 mV
La temperatura media es: 2415 (unidades en 0,01°C)
La intensidad es: 1745 mA
El estado de carga de la batería es de: 2133 (unidades en tanto por diez mil)
-----
El voltaje total es: 37560 mV
La temperatura media es: 2480 (unidades en 0,01°C)
La intensidad es: 1690 mA
El estado de carga de la batería es de: 2129 (unidades en tanto por diez mil)
-----
El voltaje total es: 37553 mV
La temperatura media es: 2350 (unidades en 0,01°C)
La intensidad es: 1717 mA
El estado de carga de la batería es de: 2125 (unidades en tanto por diez mil)

```

*Figura 46 Captura del Hercules sobre la evolución del SoC*

Se puede observar en Figura 46 que la capacidad total perdida en este intervalo de tiempo de 75s es de 0,19% descargando a una intensidad media de 1,76 A según la medida del microcontrolador. Considerando la medida de la pinza de 1,78 A constantes, se obtiene que la capacidad perdida en ese intervalo de tiempo es de 37,08 mAh. Haciendo la conversión al estado de carga se obtiene que el porcentaje descargado en dicho intervalo es de 0,37%.

Como conclusión se puede decir que la estimación del Coulomb Counting no acaba de ser exacta. El error en la medida de corriente y el posible error en el *timer* del microcontrolador hacen que el error sea considerable en tan solo 75s.

## 6.7. Prueba de descarga

Se hizo una descarga parcial de 15 minutos a la batería y a partir de los datos recibidos por Hercules se han hecho varias gráficas. La primera gráfica corresponde a la evolución del voltaje y la corriente.



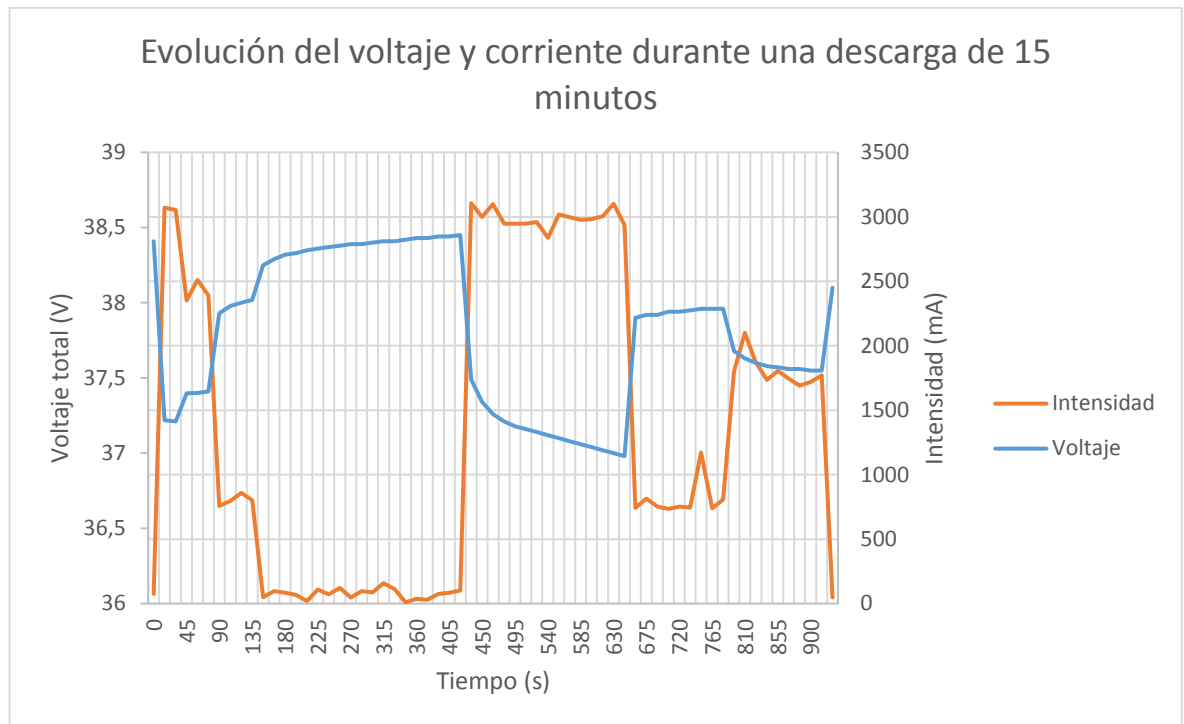


Figura 47 Evolución del voltaje y corriente durante una descarga de 15 minutos. Descargas a C/3, C/12,5 y C/5,5

La Figura 47 se ha obtenido después de un ensayo de descarga. Se observa que el voltaje leído está relacionado con la intensidad a la que se descarga. Esto es debido a que la batería no es como una fuente de tensión ideal de impedancia cero, sino que tiene impedancia y capacitancia. Por ese motivo cuando no hay intensidad durante el intervalo de 135 s a 405 s se puede observar como el voltaje va ligeramente subiendo debido a los procesos internos de difusión iónica de las celdas que componen la batería. Por el contrario, cuando la intensidad es máxima durante el intervalo de 450 s a 675 s se ve como el voltaje de la batería va bajando.

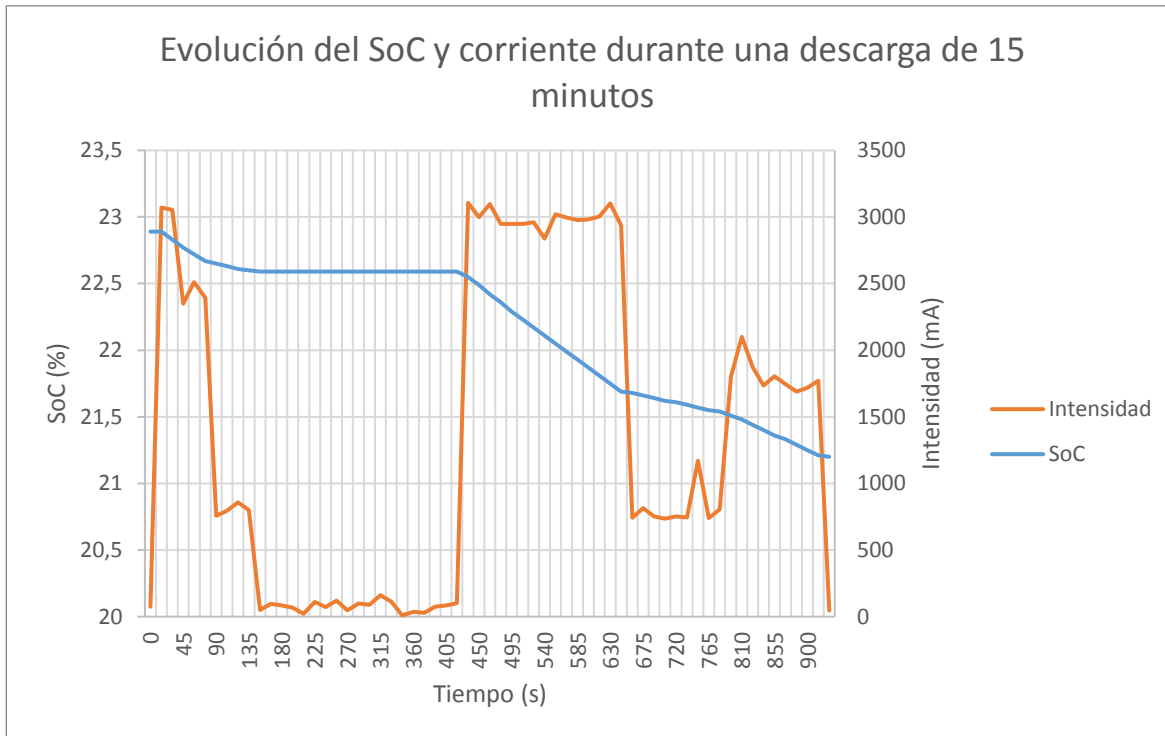


Figura 48 Evolución del SoC y corriente durante una descarga de 15 minutos. Descargas a C/3, C/12,5 y C/5,5

En la Figura 48 se observa la evolución del SoC y de la corriente durante la descarga. Aunque la estimación del SoC no sea del todo exacta, se puede observar que se comporta como debería durante la descarga. La pendiente es más o menos pronunciada según la intensidad que se lee. También se observa que en el intervalo de 135 s a 405 s donde se asume que no hay intensidad, el SoC es constante y no entra a funcionar el OCV porque no pasa el tiempo suficiente como para considerar que la batería está en reposo.

## 6.8. Comparación de la impedancia de una celda a diferentes temperaturas

Aunque este trabajo no abarca profundamente el tema de la impedancia de las celdas, se expone los test que se han hecho. Con la celda totalmente cargada y aprovechando que después de todos los ensayos del OCV la cámara térmica estaba a 60°C, se ha hecho un test de impedancia a la celda a dicha temperatura. Posteriormente cada vez que se enfriaba 10°C se repetía el test hasta que en vez de hacer el de 30°C se hizo el de temperatura ambiente (25°C).

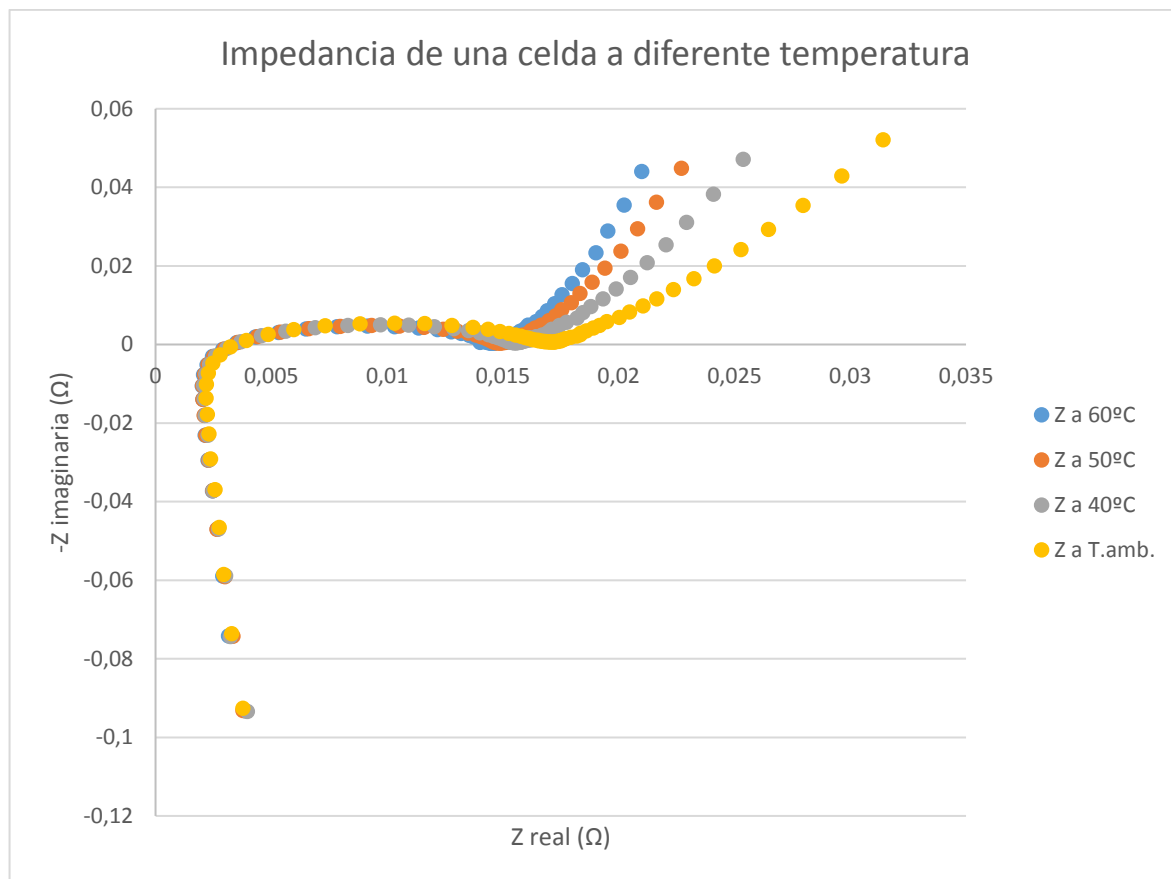


Figura 49 Comparación de la impedancia de una celda a diferentes temperaturas

La primera parte del comportamiento corresponde, como se ha dicho anteriormente, a la componente inductiva del cableado y conexionado. El paso por el 0 de la Z imaginaria, marcaría la resistencia teórica en serie que presenta la celda. Se puede observar que a todas las temperaturas se pasa, más o menos, por el mismo punto. Más o menos todas siguen la misma línea hasta que se empiezan a separar cuando la Z real alcanza un valor de 10 m $\Omega$ . Se puede observar que, a partir de este punto, cuanto más alta es la temperatura más pequeña es la impedancia que presenta.

## 6.9. Envejecimiento de la celda con la realización de los test del OCV

Este apartado es una consecuencia del anterior. El hecho de tener graficadas la impedancia antes de hacer los ensayos de carga y descarga y de tener la gráfica posterior de la impedancia, hace casi obligatorio la comparación de ambas gráficas y la comprobación del envejecimiento de la celda.

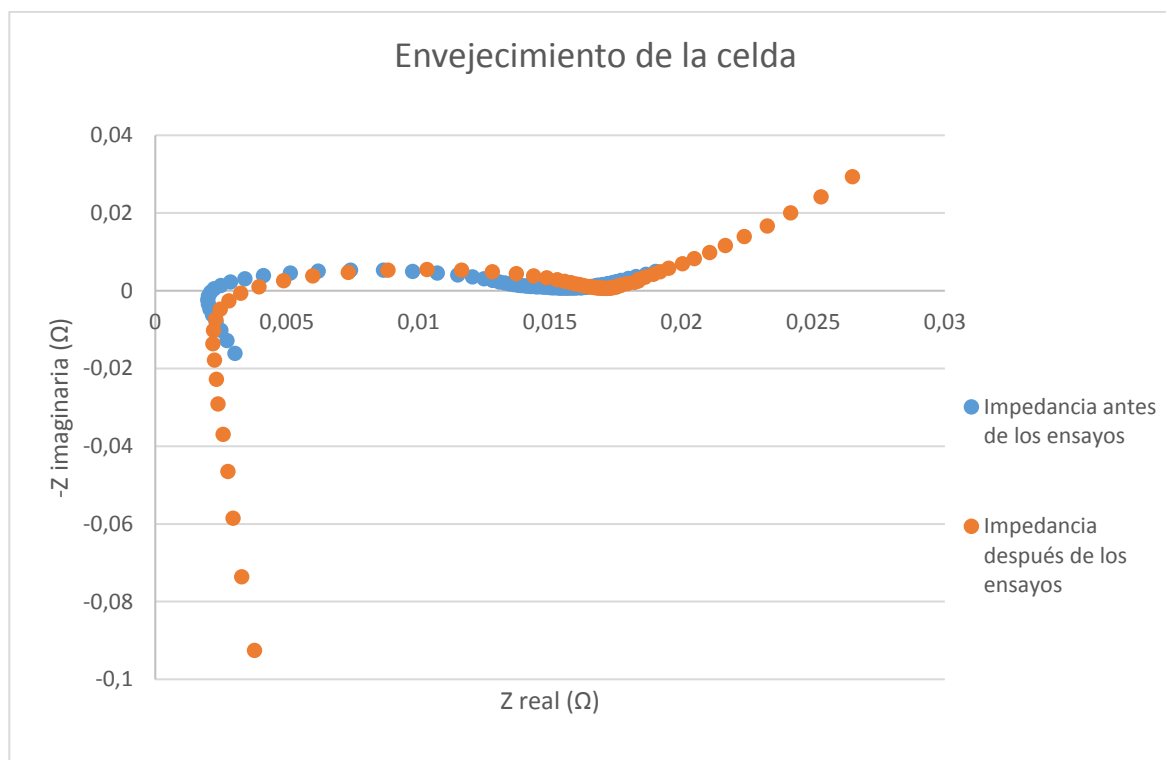


Figura 50 Envejecimiento de las celdas

En la Figura 50 se ve las gráficas resultantes de dicha comparación. Dejando de lado la primera parte que corresponde al comportamiento inductivo del cableado, se puede observar que la resistencia en serie teórica (que corresponde a la  $Z_{\text{real}}$  cuando la  $Z_{\text{imaginaria}}$  es cero) se ha reducido pero que la componente imaginaria ha crecido notablemente aumentando consigo la impedancia total de la celda a bajas frecuencias.

## 7. Análisis del impacto ambiental

Este capítulo abarca todo el impacto ambiental que provoca o puede provocar este trabajo. Se empieza comentando el efecto de la fabricación. Después, se plantea el impacto que tiene el BMS en su funcionamiento normal. Finalmente, se explica cómo se reciclan los materiales.

Todo componente que se ha utilizado para la realización del BMS cumple con la normativa RoHS (Restriction of Hazardous Substances) ya que es obligatorio desde el 2006. Esta normativa consiste en la restricción de ciertas sustancias peligrosas en aparatos eléctricos y electrónicos. Por este motivo se puede afirmar que el BMS no contiene ninguno de los seis elementos prohibidos: plomo, mercurio, cadmio, como VI, PBB y PBDE.

Como el alcance de este proyecto es diseñar y fabricar un BMS para un monoplace de Formula Student eléctrico, el impacto ambiental que tendrá en su funcionamiento normal será mínimo ya que la movilidad eléctrica no genera gases contaminantes ni excesivo ruido como para provocar contaminación acústica. Además, el hecho de investigar sobre temas relacionados con la movilidad sostenible como lo es el BMS, favorece tanto el cambio de la movilidad de combustible fósil a la movilidad eléctrica y sostenible como el cambio de modelo energético. Un buen BMS gestiona la batería de manera que favorece la eficiencia energética de ésta. Este hecho hace que se aproveche mejor la energía de la batería y, en consecuencia, que se contamine menos.

En el caso de que ya no se utilice más el BMS o la batería, ya sea por un mal funcionamiento, por el desuso o por la substitución del sistema se pueden reciclar ambos elementos. Los componentes electrónicos al ser RoSH *compliant* no tienen elementos excesivamente contaminantes a parte del plástico de sus encapsulados, aunque se pueden reciclar reutilizándolos en otros sistemas o en evoluciones del mismo o, incluso, separando las materias primas para fabricar otros componentes. Las baterías al tener componentes de Litio son más contaminantes y el hecho de que hayan perdido muchos ciclos de vida después de sus usos hace que no sean útiles para ser reutilizadas. Aunque, hoy en día, el reciclaje de dichas baterías no está excesivamente extendido y no es una práctica muy utilizada, sí que existe la posibilidad de reciclar la batería y reutilizar los componentes químicos que la forman para fabricar otra batería.



## 8. Futuros estudios

Este apartado analiza los resultados obtenidos con este trabajo, propone formas de mejorarlo y define las posibles formas de adaptar el BMS realizado para hacerlo modular para ser compatible con baterías que tengan un mayor número de celdas. Además, se proponen ideas que se pueden llegar a implementar para hacer un BMS más completo y competitivo con los que hay en el mercado.

### 8.1. Mejoras en las medidas

Vistos los resultados obtenidos, se puede afirmar que se debería mejorar la precisión y la exactitud de dos de las tres medidas realizadas. Por una parte, la medida de voltaje, como ya se ha expresado varias veces, la hice un circuito integrado de Linear Technology y los resultados obtenidos son más que aceptables, con errores inferiores a los 2 mV comparando los resultados con multímetros de gran calidad. Por otra parte, las medidas de temperatura de temperatura e intensidad han sido un poco más inexactas.

En cuanto a la medida de corriente, sí que se puede considerar mínimamente precisa ya que los resultados obtenidos eran estables y no tenían excesiva variación, pero sí que les faltaba exactitud ya que tenían más o menos 1°C de diferencia comparándolo con la medida de otros dispositivos. Ese error puede venir dado por muchos factores que en cierta medida no se pueden controlar como serían las tolerancias, las conexiones, el cableado o las interferencias electromagnéticas, entre otras cosas. Por lo que en un futuro lo que se puede hacer, es elegir NTCs y resistencias con tolerancias más ajustadas, intentar mejorar las conexiones, utilizar solo el cableado necesario e incluso añadir un condensar antes de la entrada del ADC para reducir mínimamente el ruido. Otro tema es que, si se linealiza la NTC hay otro factor que genera error de exactitud, por lo que tal vez es mejor no linealizar y mediante una *look-up table* previamente calculada, relacionar directamente el valor leído por el ADC con una temperatura. De esta manera, además de ganar exactitud, se podría incluso a llegar a optimizar el programa, haciéndolo que ocupe menos memoria y sea más rápido.

La medida de corrientes mejoró con el procesamiento software, pero a la hora de que se necesite cambiar de sensor y medir corrientes mayores el error en precisión y exactitud se hará más grande. Al tener que medir corrientes más grandes, la sensibilidad del sensor hall tendrá que ser mayor por lo que posibles interferencias o pequeños errores en la medida del ADC pueden provocar grandes errores en la medida final de la corriente. Una forma de darle una pequeña estabilidad a la señal es pasar el voltaje que se tiene que medir por un filtro pasa-bajos, de manera que eliminase posibles pequeñas oscilaciones en el voltaje que lee el ADC sacrificando las lecturas de posibles picos de corriente.

## 8.2. Mejoras de la estimación del SoC

Dado que el SoC se estimaba a partir de dos métodos, para mejorarlo se debería mejorar dichos métodos. En lo que se concierne al OCV, se deberían repetir los ensayos reduciendo la corriente de carga o de descarga para que se asemeje más al circuito en voltaje abierto para hacerlo más exacto. O incluso, hacerlo aún más exacto haciendo ensayos de carga y descarga mediante pulsos. Esta forma de carga y descarga mediante pulsos se basa en cargar/descargar a corriente constante durante un pequeño período de tiempo de manera que, sabiendo la capacidad total, la intensidad y el tiempo se sabe el porcentaje de batería que se ha quitado/aportado y el voltaje se mediría completamente en circuito abierto después de que la celda haya reposado de dicho pulso. El problema de este método es que requiere de mucho tiempo. Existen dos formas para mejorar el Coulomb Counting. La primera tiene que ver con una buena medida de corriente que ya se ha comentado en el apartado anterior. La segunda es mejorando la integración. Debido a que esta operación se hace mediante un microcontrolador se considera la integral como un sumatorio de intervalos. Por lo tanto, para mejorar el resultado es necesario reducir la amplitud de dichos intervalos para que el sumatorio de un resultado más parecido al de la integración real.

## 8.3. Ampliar el sistema para un mayor número de celdas

Para conseguir que este BMS sea capaz de funcionar para batería que tengan más de 12 celdas se tendrían que hacer diversas modificaciones. Dichas modificaciones afectarían tanto al hardware como al software. La topología del BMS pasaría de ser centralizada a ser modular.

Primero, para simplificar cableado, facilitar el manejo del módulo y reducir costes sería necesario unificar las dos PCBs que forman el BMS de este trabajo. De esta manera se tendría un módulo que constaría de una sola PCB cada 12 celdas. Además, no sería necesario que todos los módulos fuesen capaces de medir la corriente (ya que es la misma para todo el conjunto) ni que fuesen capaces de activar o desactivar un relé. Solo sería necesario que estas dos funciones las hiciera un solo módulo, que sería el encargado de recopilar la información de los demás módulos, de estimar el SoC, de decidir si el estado de la batería es correcto o no como para activar o desactivar el relé y de comunicarse con el resto de electrónica del coche. También sería necesario un convertidor DC/DC aislado de la entrada, ya que el GND de la alimentación de la PCB se cortocircuita con el polo más negativo de la batería. Por el mismo motivo el CAN *transceiver* también debería ser aislado.

En resumen, sería necesario un módulo que fuese capaz de controlar dos líneas de CAN: una para la comunicación entre módulos y para comunicarse con el resto de la electrónica del coche, que midiese la corriente, estimase el SoC y determinase el estado de la batería. Los demás módulos solo deberán



medir voltajes y temperaturas y actuar al respecto balanceando celdas o avisando que algún parámetro sale del límite.

## 8.4. Posibles mejoras adicionales

Una de las posibles mejoras consiste en añadir una tarjeta SD para guardar información sobre los diferentes parámetros de la batería durante su carga, descarga o reposo. De esta manera se podrían obtener datos de la batería en pleno funcionamiento que podrían ayudar a entender de una mejor manera el comportamiento de las celdas que la componen. Para añadir una tarjeta SD, aparte de entender cómo funcionan, se necesitaría utilizar un microcontrolador que soportase dos líneas de comunicación SPI, ya que tendría que controlar tanto el LTC6811-2 como la tarjeta SD.

Otra posible mejora es hacer una aplicación para el ordenador para visualizar de una forma más ordenada los datos recibidos por RS232. Además de visualizar los datos, podría graficarlos de manera que se tuviese constancia la evolución de los valores de voltaje, intensidad, temperatura y estado de carga de una forma visual y fácil de entender.

Un sistema de detección de fallos también sería una mejora significativa. Este sistema sería capaz de detectar si falla algún componente, si la alimentación es correcta o si las comunicaciones están funcionando correctamente. Para llevarlo a cabo se necesitarían monitorizar muchas señales digitales por el microcontrolador, por lo que se necesitaría un microcontrolador con más pines de entrada que el que se ha utilizado para este trabajo. A parte de la detección de fallos, se podría añadir un *watchdog* para evitar que el software entre en posibles bucles. De esta manera se aseguraría el correcto funcionamiento del sistema, y en caso de fallo el propio BMS lo detectaría por lo que sería simplemente reparar el componente/es dañados.



## Conclusiones

El objetivo principal era el diseño de un BMS funcional. Se puede considerar cumplido ya que dicho sistema hace las medidas correctamente, hace correctamente el balanceo y permite tanto carga como descarga. El BMS se ha hecho para una batería de 12 celdas de  $\text{LiFePO}_4$  con la que, gracias al sistema de descarga, se ha comprobado el correcto funcionamiento de éste.

El estado de carga se decidió hacerlo por combinación de dos métodos: el OCV y el Coulomb Counting. Se puede afirmar que el OCV (que está en función de la temperatura) funciona correctamente pero el Coulomb Counting no ha dado el resultado esperado.

Las comunicaciones se han realizado de manera satisfactoria. El BMS envía correctamente los datos obtenidos tanto a un ordenador externo vía RS232 como a otro microcontrolador vía CAN.

El cuarto objetivo también se ha cumplido. Se ha diseñado y fabricado un sistema que sirve para descargar la batería a través de varias resistencias de potencia para poder testear de manera correcta el BMS.

El último objetivo era marcar las bases para mejorar el trabajo realizado. Se puede considerar cumplido ya que, después de los errores cometidos y los conocimientos obtenidos a partir de la realización de este trabajo, se realizará, en la temporada 2017-2018, el BMS para la batería del equipo de Formula Student, e-Tech Racing.

Como conclusiones generales se puede decir que este trabajo ha hecho necesaria la aplicación de los conocimientos adquiridos durante el grado de ingeniería electrónica industrial y automática, así como ampliar conocimientos relacionados con el funcionamiento y comportamiento de celdas de Li-ion, los microcontroladores PIC o los protocolos de comunicación SPI, CAN y RS232, entre otros.



## Presupuesto y análisis económico

Este apartado incluye con todo detalle el coste del trabajo. Se analiza tanto el precio de los materiales como las horas de trabajo como el precio de los programas utilizados. Para hacerlo de una manera más ordenada y entendible se explicará por partes y, finalmente, se hará el resumen total.

Primero de todo, se muestra el coste de los softwares necesarios para realizar este proyecto. La Tabla 3 tiene dichos datos. Cabe destacar que solo se han nombrado los programas que se han utilizado de forma continuada, no se han tenido en cuenta los programas que se han usado de forma esporádica.

Tabla 3 Costes de los softwares utilizados

Softwares	
Nombre	Precio
TARGET3001	0€ (versión gratuita)
PIC C Compiler	350€
Microsoft Office	150€
PICkit3	50€
Hercules	0€
<b>Precio total</b>	<b>550€</b>

Al comienzo del trabajo se estuvo mucho tiempo de estudio, aprendizaje y diseño. La Tabla 4 muestra el precio de forma detallada de todo este proceso.

Tabla 4 Costes del estudio del sistema y su diseño

Estudio y diseño			
Acción	Horas	Precio por hora	Precio

Estudio y entendimiento del funcionamiento de un BMS y de sus características	30h	10€/h	300€
Decisión del tipo de BMS y de batería	20h	15€/h	300€
Pruebas de diferentes hardware en protoboard	100h	15€/h	1500€
Diseño final del hardware	45h	15€/h	675€
<b>Precio total</b>			<b>2775€</b>

Posteriormente, se tuvo que comprar todos los materiales, soldar los componentes, hacer el cableado. Básicamente se tuvo que montar todo el sistema junto con la batería para hacer las pruebas. La Tabla 5 contiene el precio de todo este proceso y se divide en dos partes: la parte de materiales y componentes comprados y la parte de fabricación, soldadura, cableado y montaje. Los precios de la parte de materiales provienen de la web del fabricante o RS-online que es un distribuidor de componentes electrónicos.

Tabla 5 Costes de los materiales y la fabricación del trabajo

Materiales y componentes			
Material	Unidades	Precio por unidad	Precio
Batería	1	340€	340€
Demoboard DC2260A de Linear Technology	1	150€	150€
PCB	5 (mínimo)	11,6€	58€
Resistencia SMD	40 (mínimo)	0,1€	4€

Multiplexor	10 (mínimo)	0,56€	5,6€
NTCs	15 (mínimo)	0,21€	3,2€
Referencia de tensión	1	7,5€	7,5€
Amplificador operacional Rail to Rail	1	4,4€	4,4€
Condensadores	15 (mínimo)	0,1€	1,5€
Cristal de cuarzo	5 (mínimo)	0,42€	2,1€
LEDs	2	1,6€	3,2€
Conectores	5 unidades de 20 pines	0,74€	3,7€
Microcontrolador	1	7,92€	7,92€
CAN transceiver	1	0,95€	0,95€
Sensor Hall	1	15,65€	15,65€
Cable	100 metros (mínimo)	11,95€	11,95€
Resistencias de potencia	3	5,6€	16,8€
Interruptores	3	2,2€	6,6€
Transistores	5 (mínimo)	0,33€	1,65€
MOSFET	10 (mínimo)	0,53€	5,3€
<b>Precio total</b>			<b>650€</b>
<b>Fabricación</b>			
<b>Acción</b>	<b>Horas</b>	<b>Precio por hora</b>	<b>Precio</b>
Preparación de la batería	3h	10€/h	30€

Soldar	5h	10€/h	50€
Cableado	18h	10€/h	180€
Montaje final	5h	10€/h	50€
<b>Precio total</b>			<b>310€</b>

Aunque las bases de la programación ya se empezaron a plantear durante el diseño el hardware, ahora que se tenía tanto la PCB como y el montaje final era el momento de hacer los programas para probar los diferentes elementos y definir el programa final. La Tabla 6 muestra las horas y las actividades que se hicieron en este proceso.

*Tabla 6 Costes del proceso de programación y testeo*

<b>Programación y testeo</b>			
<b>Acción</b>	<b>Horas</b>	<b>Precio por hora</b>	<b>Precio</b>
Programación y testeo de la medida de temperatura	8h	15€/h	120€
Programación y testeo de la medida de corriente	10h	15€/h	150€
Programación y testeo de la medida de voltaje y balanceo	15h	15€/h	225€
Unificación programa de medidas y testeo	15h	15€/h	225€
Programación y testeo de la estimación del SoC	25h	15€/h	375€
Programa final	30h	15€/h	450€



<b>Precio final</b>	<b>1545€</b>
---------------------	--------------

La última fase del proyecto ha sido la redacción de la memoria. Esta fase no tiene tabla en si porque es una sola actividad. Directamente aparece reflejada en el resumen final, teniendo en cuenta que se ha tardado 100 horas en redactarla y el precio por hora era de 15€/h.

Finalmente, y una vez definidos las diferentes fases se ha hecho una tabla resumen (Tabla 7) del precio de cada uno de ellas y el precio total. Cabe destacar que un dato que no aparece en la tabla es las horas de dedicación totales, que si sumamos las diferentes fases sale un total de 429 horas.

Tabla 7 Resumen de los costes del trabajo

<b>Proceso</b>	<b>Precio</b>
Softwares	550€
Estudio y diseño	2775€
Materiales y componentes	650€
Fabricación	310€
Programación y testeo	1545€
Redacción de la memoria	1500€
<b>Precio total</b>	<b>7330€</b>



## Bibliografía

- [1] “Coche eléctrico.” [Online]. Available: [https://es.wikipedia.org/wiki/Automóvil\\_eléctrico](https://es.wikipedia.org/wiki/Automóvil_eléctrico). [Accessed: 26-Apr-2017].
- [2] “Diferencias entre coches híbridos y eléctricos.” [Online]. Available: <http://www.autobild.es/contenido-patrocinado/hibrido-hibrido-enchufable-electrico-sus-diferencias-244147>. [Accessed: 26-Apr-2017].
- [3] “Ventajas e inconvenientes del coche eléctrico.” [Online]. Available: <http://coches1a.es/ventajas-desventajas-de-los-coches-electricos/>. [Accessed: 26-Apr-2017].
- [4] “Evolución de las baterías en el vehículo eléctrico.” [Online]. Available: <https://www.xataka.com/automovil/asi-han-progresado-las-baterias-de-los-coches-electricos-12-veces-mas-autonomia-en-100-anos>. [Accessed: 26-Apr-2017].
- [5] “Comparación tipos de celdas.” [Online]. Available: [http://batteryuniversity.com/learn/article/secondary\\_batteries](http://batteryuniversity.com/learn/article/secondary_batteries). [Accessed: 26-Apr-2017].
- [6] “Baterías de NiFe.” [Online]. Available: [http://batteryuniversity.com/learn/article/nickel\\_based\\_batteries](http://batteryuniversity.com/learn/article/nickel_based_batteries). [Accessed: 26-Apr-2017].
- [7] “Diagrama de Ragone.” [Online]. Available: <http://www.silverbearcafe.com/private/09.12/batteries.html>. [Accessed: 27-Apr-2017].
- [8] “Tipos de celdas de Litio.” [Online]. Available: [http://batteryuniversity.com/learn/article/types\\_of\\_lithium\\_ion](http://batteryuniversity.com/learn/article/types_of_lithium_ion). [Accessed: 27-Apr-2017].
- [9] B. Santos, “Battery Management System applied to Proyecto FST ’ s EV prototype” p.8, p16, p.17, p.18, no. October, 2014.
- [10] J. Sainio, “Battery Management System Design and Implementation in Electric Raceabout - Electric Sportscar” p.57, no. May, 2013.
- [11] W. S. Iii, “Lithium-based battery system management and balancing”p.8, 2011.
- [12] “Graficas de una celda de Li-ion.” [Online]. Available: [http://www.richtek.com/en/Design\\_Support/Technical\\_Document/AN024](http://www.richtek.com/en/Design_Support/Technical_Document/AN024). [Accessed: 27-Apr-2017].
- [13] Y. M. Jeong, Y. K. Cho, J. H. Ahn, S. H. Ryu, and B. K. Lee, “Enhanced coulomb counting method with adaptive SOC reset time for estimating OCV,” *2014 IEEE Energy Convers. Congr. Expo. ECCE 2014*, pp. 4313–4318, 2014.
- [14] Z. Aungkyaw, H. Lahiji, M. Riesmeyer, and A. Supervisor, “Lithium-Ion Battery Management System for FSAE Electric Vehicles,” pp. 0-90, no. March, 2016.
- [15] H. Fisk and J. Leijgård, “A Battery Management Unit,” no. June, p. 60, 2010.

- [16] "Batería." [Online]. Available: <https://www.goldenmotor.com/>. [Accessed: 29-Apr-2017].
- [17] "Datasheet de LTC6811." [Online]. Available: <http://cds.linear.com/docs/en/datasheet/68111f.pdf>. [Accessed: 29-Apr-2017].
- [18] "Esquemático de DC2260A-2." [Online]. Available: [http://cds.linear.com/docs/en/demo-board-schematic/DC2260A-2\\_SCH.PDF](http://cds.linear.com/docs/en/demo-board-schematic/DC2260A-2_SCH.PDF). [Accessed: 28-Apr-2017].
- [19] D. de ingeniería electrónica EUETIB UPC, "" Instrumentació Electrònica ' Termistores NTC."
- [20] "Datasheet de LTC6201." [Online]. Available: <http://cds.linear.com/docs/en/datasheet/62001ff.pdf>. [Accessed: 04-May-2017].
- [21] "Datasheet de LTC1019." [Online]. Available: <http://cds.linear.com/docs/en/datasheet/1019fd.pdf>. [Accessed: 04-May-2017].
- [22] "Datasheet de PIC18F2685." [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/39761b.pdf>. [Accessed: 04-May-2017].



## TRABAJO FINAL DE GRADO

**Grado en Ingeniería Electrónica Industrial y Automática**

# **BATTERY MANAGEMENT SYSTEM (BMS) PARA UN COCHE DE FORMULA STUDENT**



## **Volumen II**

### **Anexos**

**Autor:** Lucas Bouzón Pousa  
**Director:** Ángel Cuadras  
**Convocatoria:** Junio 2017



## Anexo A

En este anexo se añade la información más técnica del trabajo que no se ha mostrado en la memoria. Esta información está compuesta por el esquemático general de la PCB diseñada, su *layout* y finalmente el código que se ha utilizado para programar el microcontrolador.

### A1. Esquemático general de la PCB diseñada

Debido a que en una sola imagen no se pueden apreciar bien los detalles, el esquemático está dividido en varias partes. Por lo que se presenta cada una de ellas y finalmente, el esquemático general. Además, se repasa el funcionamiento de la placa dando referencias al esquemático.

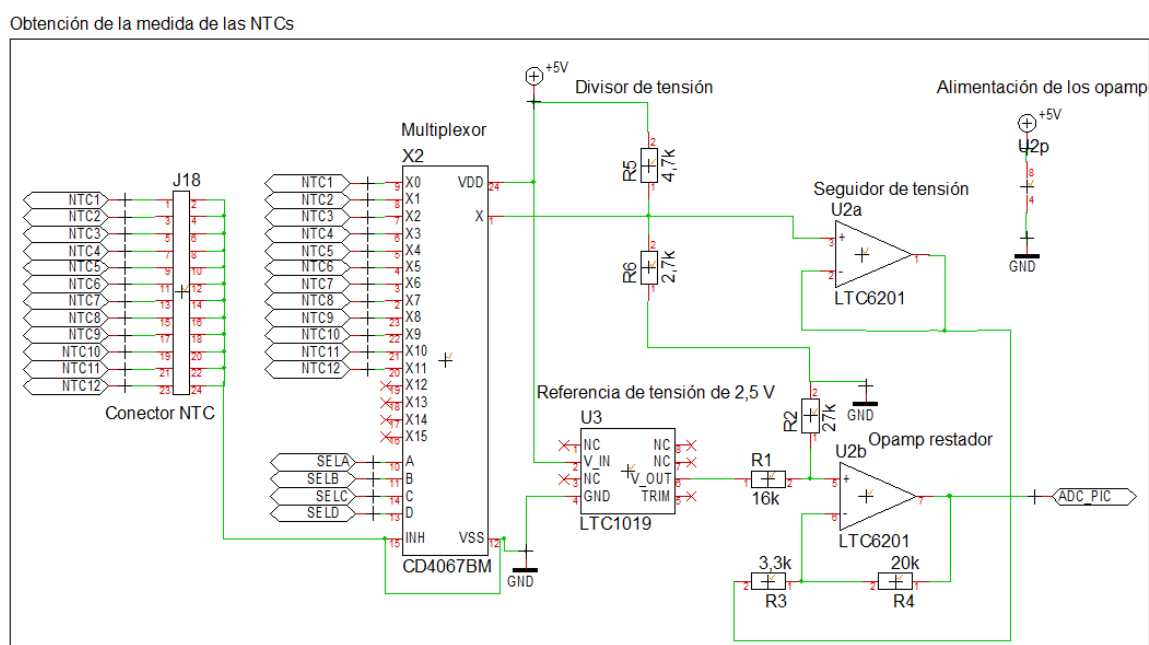


Figura 51 Esquemático de la obtención de medida de las NTCs

En la Figura 51 se muestra tanto el conector de las NTCs como la linealización de éstas a partir de las resistencias R5 y R6 como la etapa adaptadora de voltaje compuesta por los amplificadores *railtorail*, las resistencias R1, R2, R3 y R4 y por la referencia de tensión LTC1019. Aunque se haya explicado en la memoria, un breve resumen del funcionamiento del circuito es el siguiente. Se conectan las NTCs al conector, se multiplexan dichas NTCs para no tener que hacer todo el hardware posterior para cada una de ellas, se linealizan y el voltaje se adapta a partir de los componentes mencionados anteriormente para mejorar la resolución del ADC.

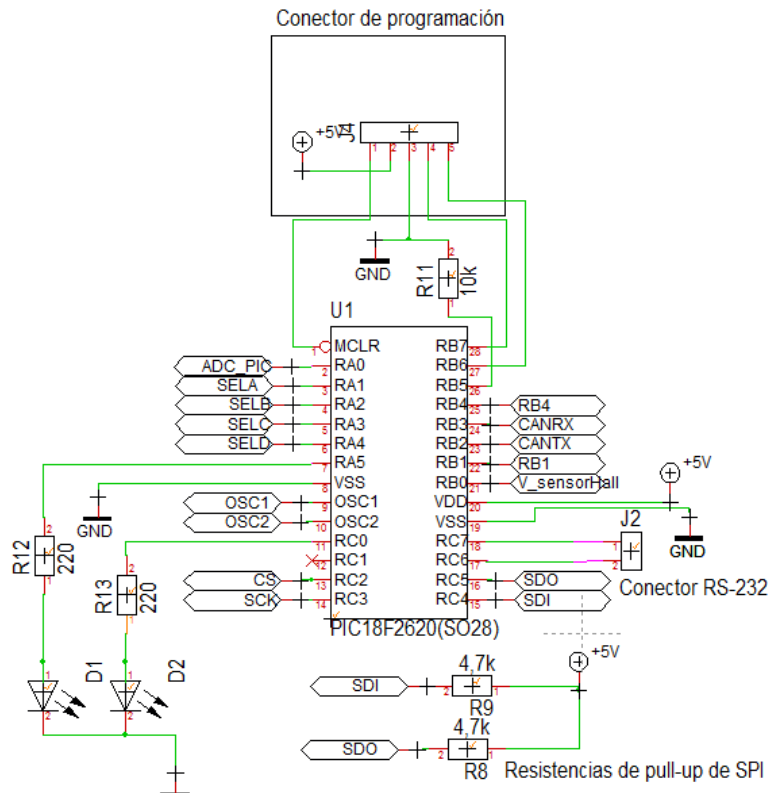


Figura 52 Esquemático del microcontrolador y sus conexiones

La Figura 52 muestra el esquemático del microcontrolador. Éste se encarga de gestionar toda la información del sistema y de tomar las decisiones. Por esa razón hace las siguientes funciones.

- Lee a través de su ADC interno las medidas de temperatura (señal: ADC\_PIC) y de corriente (señal: V\_sensorHall).
- Se encarga de seleccionar los canales del multiplexor (señales: SELA, SELB, SELC y SELD) para leer todas las temperaturas.
- Se encarga de iluminar dos LEDs a través de los puertos RA5 y RC0 según el estado de la batería.
- Se comunica vía SPI con la demoboard de Linear Technology a través de los puertos RC2, RC3, RC5 y RC4.
- Se comunica vía RS232 con un ordenador externo a través de los puertos RC6 y RC7.
- Saca dos señales digitales (señales: RB4 y RB1) que sirven para activar un relé y para simular la señal de BMS OK.
- Se comunica vía CAN a través de los puertos RB3 y RB2.
- Procesar toda la información por software.



Finalmente, destacar que tiene el conector de programación, que las señales OSC1 y OSC2 conectan con el cristal de cuarzo necesario para generar el *clock* y que las resistencias R8 y R9 son de *pull-up* para mejorar la comunicación SPI.

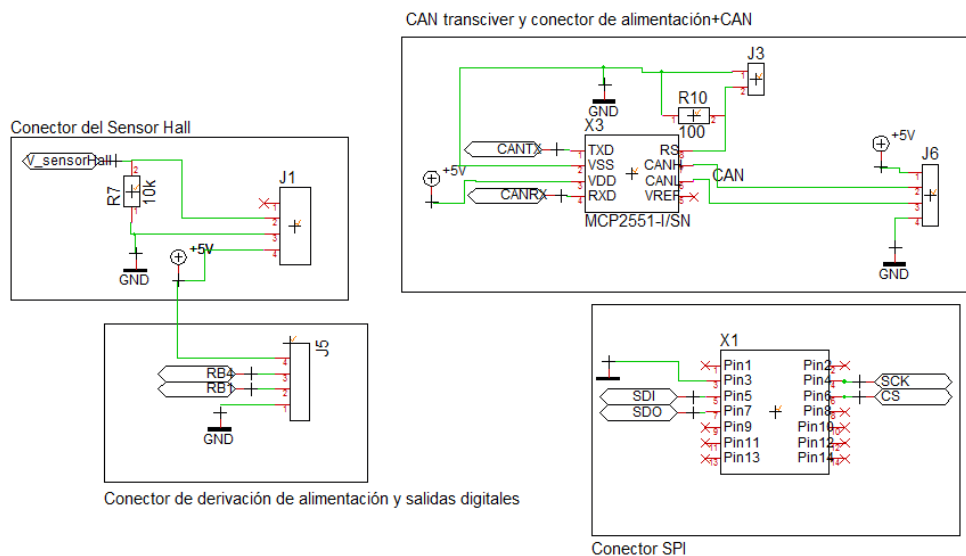


Figura 53 Conectores de la PCB

La Figura 53 muestra los conectores de la placa (exceptuando el de programación y el de las NTCs). Por una parte, se tiene el conector del sensor hall que le lleva la alimentación y recibe el voltaje de salida del sensor para leerlo. La resistencia entre la señal *V\_sensorHall* y GND es recomendada por el fabricante del sensor. Por otra parte, el conector J5 tiene las dos salidas digitales comentadas anteriormente y sirve para hacer una derivación de la alimentación o para medir la alimentación con un multímetro de manera sencilla. Antes del conector de CAN se puede observar el conexionado del CAN *transceiver* que se encarga de transformar las señales digitales del microcontrolador en señales diferenciales de CAN. Finalmente, el conector de SPI se conectará al conector SPI de la demoboard de Linear Technology y también sirve para unir el GND de la PCB con el terminal más negativo de la batería para evitar posibles errores de referencia.

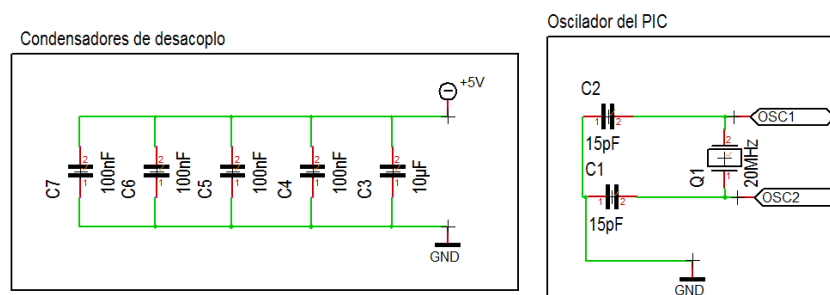


Figura 54 Condensadores de desacoplo y oscilador del microcontrolador

La Figura 54 muestra los condensadores de desacoplo y el oscilador del microcontrolador. Los condensadores de desacoplo se sitúan cerca de las alimentaciones de los circuitos integrados como pueden ser el multiplexor, los amplificadores operacionales o el microcontrolador para darle cierta estabilidad a la alimentación y evitar posibles pequeños picos de voltaje causados por el ruido. El oscilador está compuesto por un cristal de cuarzo de 20 MHz y dos condensadores de 15pF recomendados por el fabricante del microcontrolador. Finalmente, La Figura 55 muestra todo el esquemático en una sola imagen.

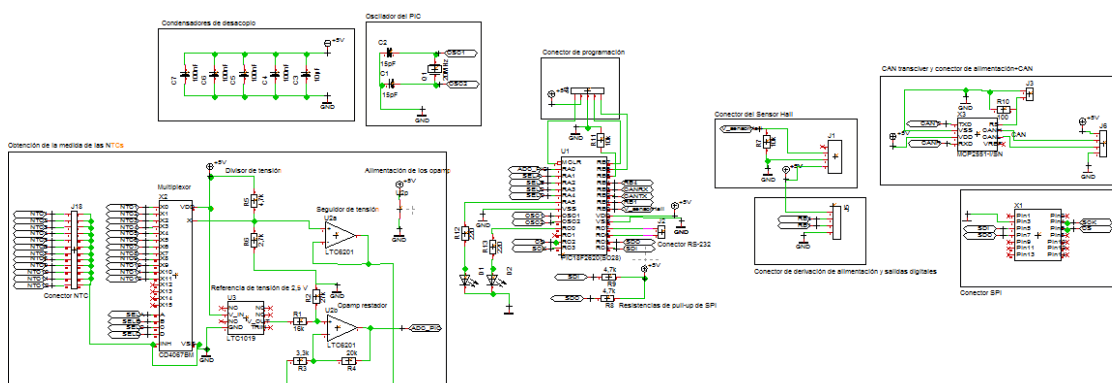
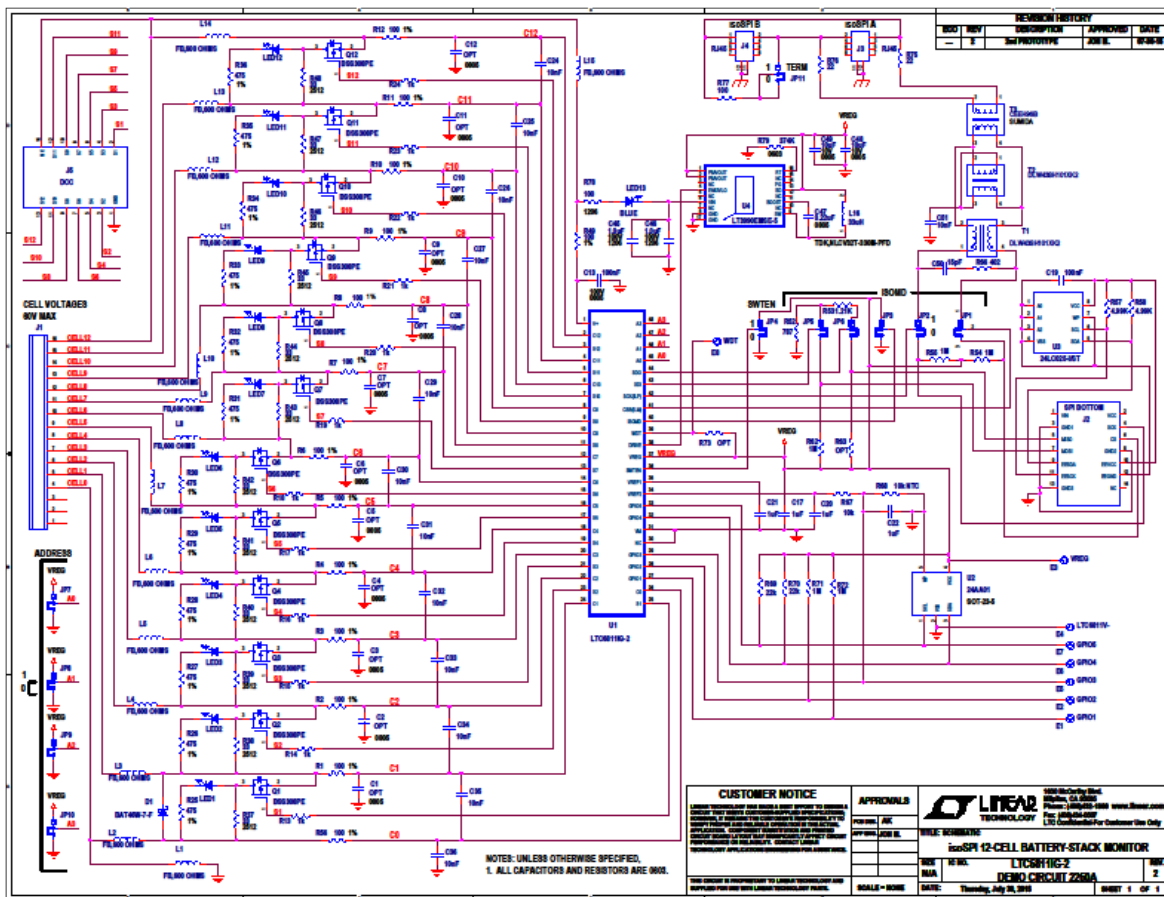


Figura 55 Esquemático general del circuito

## A2. Esquemático detallado de la placa DC2260A

Como se ha dicho en la memoria, en este apartado se expone en varias imágenes el esquemático de la demoboard de Linear Technology para que se puedan ver los detalles. Se muestran sólo las partes más relevantes del circuito para el correcto funcionamiento del BMS. La Figura 56 muestra el mismo esquemático general que aparece en la memoria.



*Figura 56 Esquemático de la DC2260A*

En la Figura 57 se pueden apreciar varios detalles. Por una parte, se ve el LTC6811-2. Por otra parte, se ve el conector de las celdas y la estructura comentada en la memoria de obtención de voltajes y balanceo. Dicha estructura se repite para todas las celdas y está compuesta por una ferrita, un filtro pasa-bajos y un condensador de desacoplo para la obtención de voltajes y por un PMOS con su resistencia en su *gate*, una resistencia de 1W, un LED y una resistencia para limitar la corriente de dicho LED.

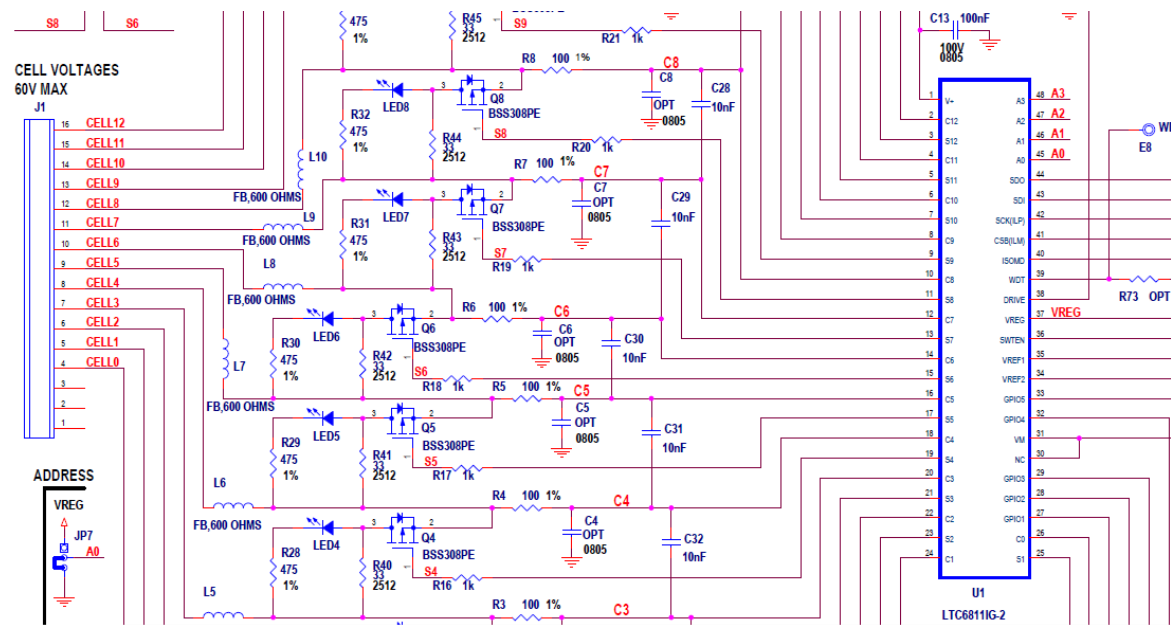


Figura 57 LTC8811-2, estructura de balanceo y obtención de voltajes y conector de las celdas de la DC2260A

La Figura 58 muestra el DC/DC (componente U4) que utiliza la demoboard para alimentar la electrónica de control del LTC6811-2 a 5 V. También muestra el hardware para el isoSPI compuesto por los transformadores T1 y T3 y los conectores J3 y J4 (en este trabajo no se utiliza esta comunicación).

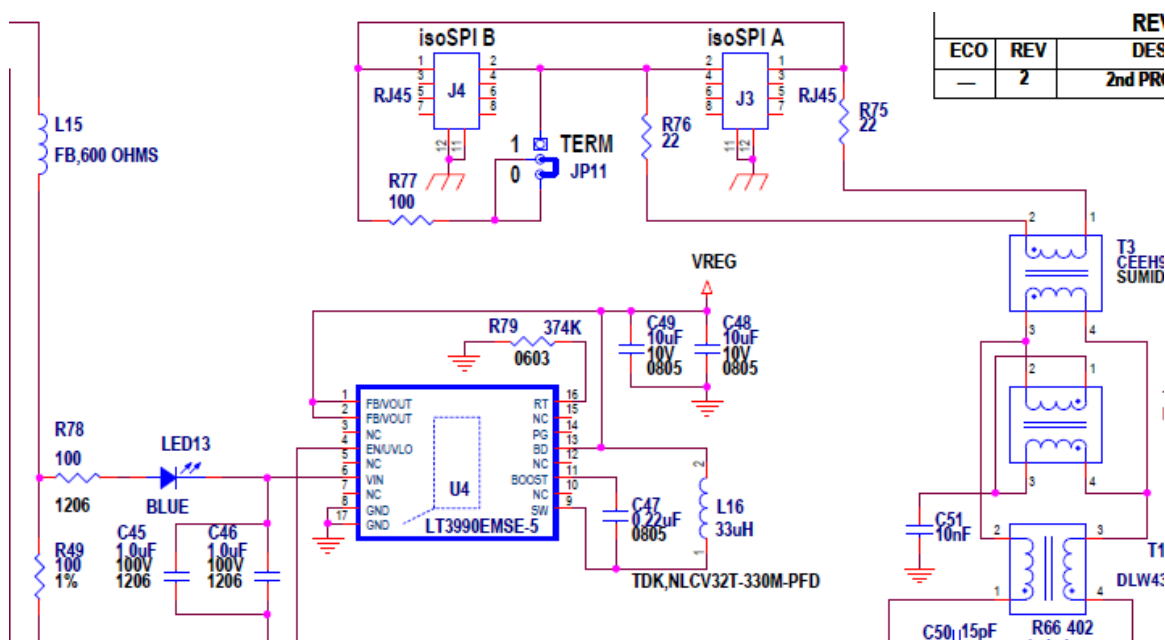


Figura 58 DC/DC y isoSPI de la DC2260A

La Figura 59 muestra los *jumpers* J1, J2, J3, J4, J5 y J6 que sirven para elegir el tipo de comunicación que se quiere utilizar para el LTC6811-2 (en el caso de este trabajo como se usa SPI se dejan todos en

la posición 0 como esta en la figura). El conector J2 tiene las señales correspondientes de SPI, I2C (no se utilizan), GND, Vin y VCC.

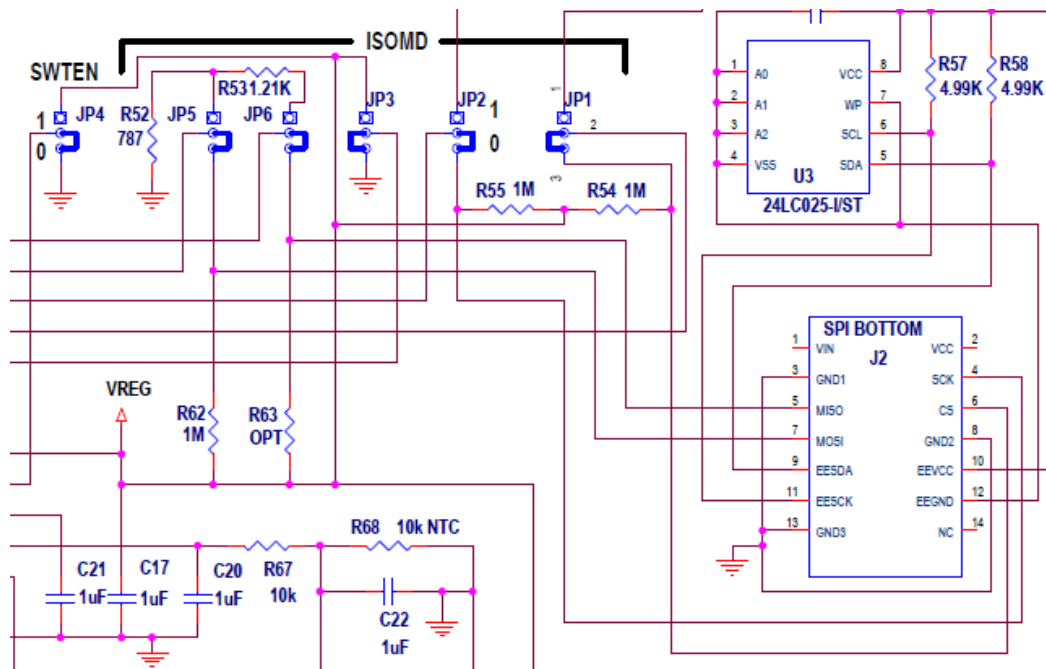


Figura 59 jumpers de selección de protocolo de comunicación y conector SPI/I2C

La Figura 60 muestra los *jumpers* que se utilizan para definir el *address* del LTC6811-2 para la comunicación. En el caso de este trabajo se ha elegido la *address* 0000 por lo que todos los *jumpers* se han dejado en la posición 0 como muestra la figura.

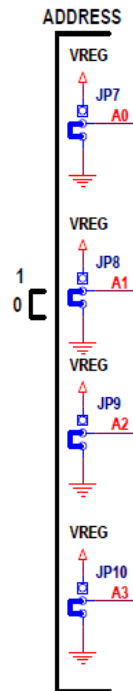


Figura 60 Selección del address del LTC6811-2

### A3. Layout de la PCB

Este apartado muestra las imágenes del *layout* realizado para fabricar la PCB. Se muestra por separado la cara *bottom* y *top* para no perder detalle de las pistas y las vías y finalmente, se muestra el layout final.

Para hacer el *layout* se han tenido en cuenta las siguientes consideraciones:

- Evitar ángulos rectos.
- El ancho de la pista debía tener un mínimo de 0,5 mm.
- El espacio mínimo entre pistas y vías debía ser de un mínimo de 0,5 mm.
- Los agujeros de las vías debían tener un diámetro de 0,6 mm.

La Figura 61 y la Figura 62 muestran las pistas y vías de la cara *bottom* y la *top*, respectivamente. La Figura 63 muestra la PCB final con todas las capas.

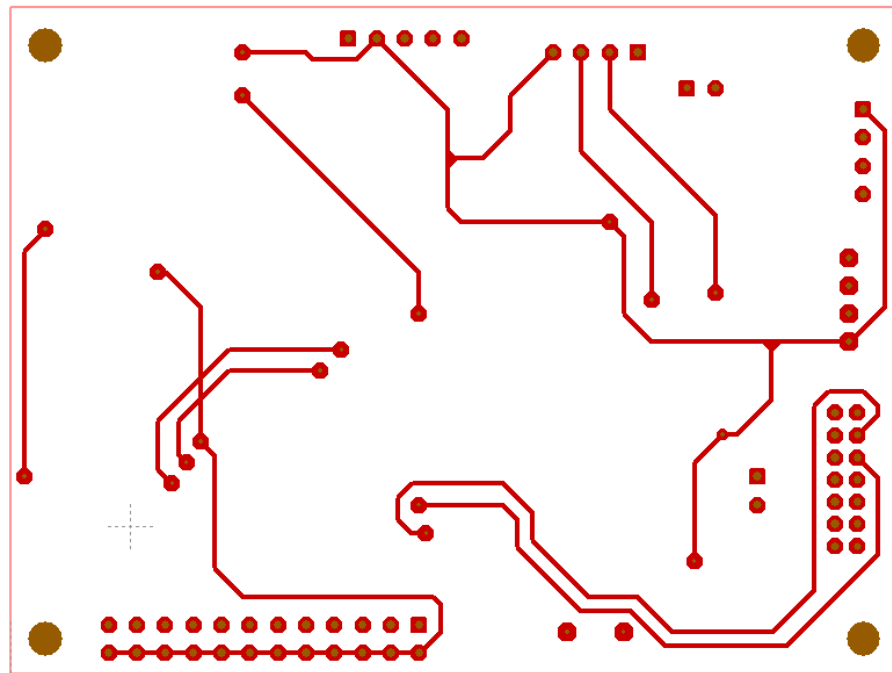


Figura 61 Layout de la cara bottom de la PCB

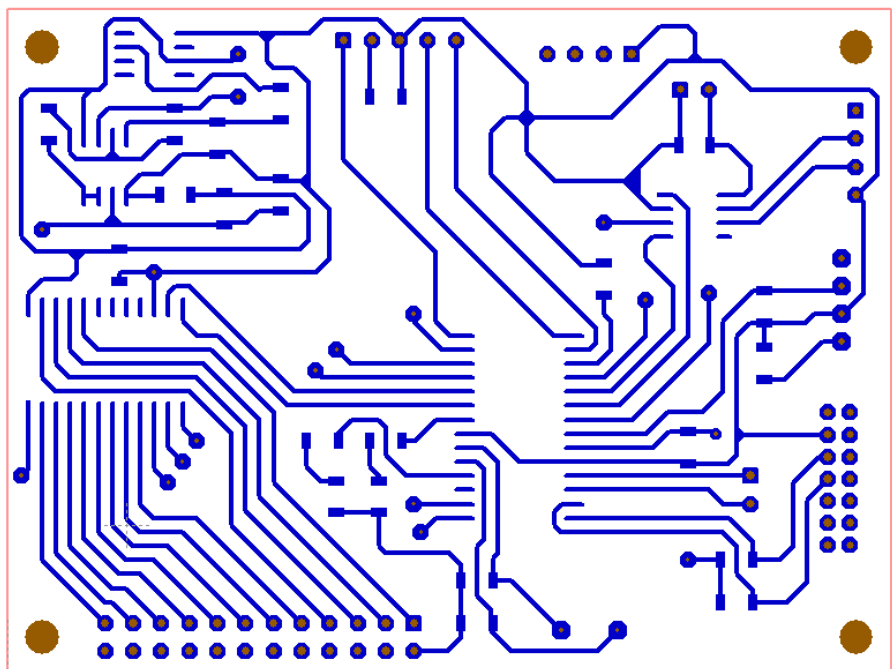


Figura 62 Layout de la cara top de la PCB

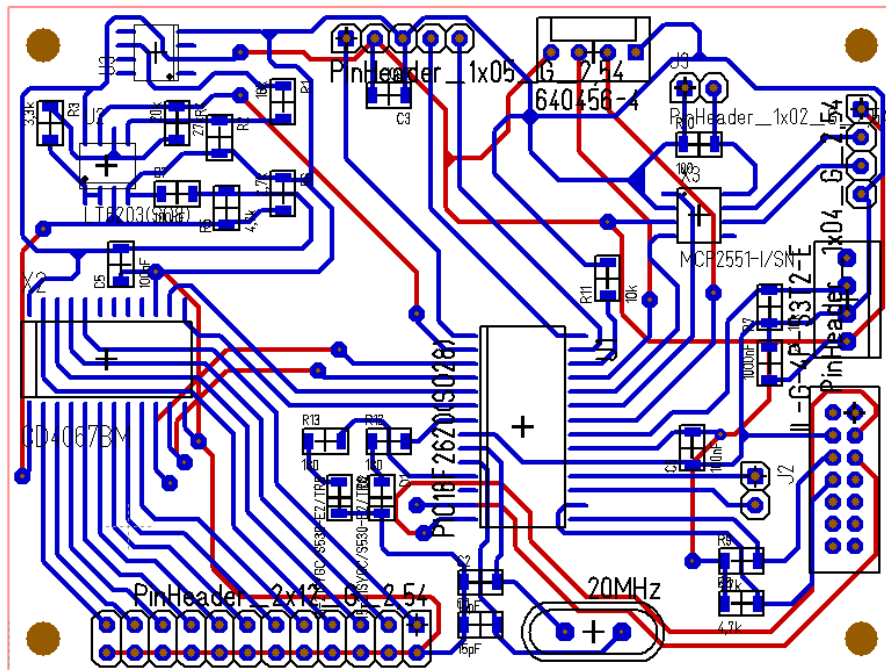


Figura 63 Layout general y final de la PCB

## A4. Programa del microcontrolador

```
#include <18F2685.h>
```

```
#FUSES NOWDT           //No Watch Dog Timer
#FUSES WDT128          //Watch Dog Timer uses 1:128 Postscale
#FUSES HS               //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES FCMEN           //Fail-safe clock monitor enabled
#FUSES IESO             //Internal External Switch Over mode enabled
#FUSES NOPUT           //No Power Up Timer
#FUSES NOBROWNOUT      //No brownout reset
#FUSES BORV21           //Brownout reset at 2.1V
#FUSES PBADEN           //PORTB pins are configured as analog input channels on RESET
#FUSES LPT1OSC          //Timer1 configured for low-power operation
#FUSES NOMCLR           //Master Clear pin used for I/O
#FUSES STVREN           //Stack full/underflow will cause reset
#FUSES NOLVP            //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES BBSIZ1K          //1K words Boot Block size
#FUSES NOXINST          //Extended set extension and Indexed Addressing mode disabled (Legacy
mode)
#FUSES NODEBUG          //No Debug mode for ICD
```



```
#FUSES NOPROTECT      //Code not protected from reading
#FUSES NOCPB          //No Boot Block code protection
#FUSES NOCPD          //No EE protection
#FUSES NOWRT          //Program memory not write protected
#FUSES NOWRTC         //configuration not registers write protected
#FUSES NOWRTB         //Boot block not write protected
#FUSES NOWRTD         //Data EEPROM not write protected
#FUSES NOEBTR         //Memory not protected from table reads
#FUSES NOEBTRB        //Boot block not protected from table reads
```

```
#use delay(xtal=20M)
```

```
//se define las comunicaciones que se usarán
```

```
#use rs232(baud=9600, xmit=PIN_C6,rcv=PIN_C7)
```

```
#use SPI (MASTER, FORCE_HW, MODE=3, BITS=8, MSB_FIRST, STREAM=SP, BAUD=9600)
```

```
#define CAN_USE_EXTENDED_ID FALSE
```

```
#include <can-18xxx8.c>
```

```
//se ponen nombres entendibles a los pines que se utilizan del PIC
```

```
#define CS pin_C2
```

```
#define LED1 pin_A5
```

```
#define LED2 pin_C0
```

```
#define RELE pin_B1
```

```
#define A pin_A1
```

```
#define B pin_A2
```

```
#define C pin_A3
```

```
#define D pin_A4
```

```
//se definen las constantes que sirvan como configuración
```

```
const int16 V_bal=3500;
```

```
const int16 V_max=3600;
```

```
const int16 V_min=2500;
```

```
const int16 I_max=6500;
```

```
const int16 Tmax=6500;
```

```
//tablas para el OCV
```

const

int16

V\_soc\_20[871]={3295,3290,3288,3286,3285,3284,3283,3282,3282,3281,3281,3281,3280,3280,3280,  
 3280,3279,3279,3279,3279,3278,3278,3278,3278,3278,3278,3277,3277,3277,3277,3277,3277,  
 3277,3277,3277,3277,3276,3276,3276,3276,3276,3276,3276,3276,3276,3276,3275,3275,3275,  
 3275,3275,3275,3275,3275,3275,3275,3275,3275,3274,3274,3274,3274,3274,3274,3274,3274,  
 3274,3274,3274,3274,3274,3274,3274,3274,3274,3274,3274,3273,3273,3273,3273,3273,  
 3273,3273,3273,3273,3273,3273,3273,3273,3273,3273,3273,3273,3273,3273,3272,3272,3272,  
 3272,3272,3272,3272,3272,3272,3272,3272,3272,3272,3272,3272,3272,3272,3272,3272,3271,  
 3271,3271,3271,3271,3271,3271,3271,3271,3271,3271,3271,3271,3271,3271,3271,3271,3271,  
 3271,3271,3270,3270,3270,3270,3270,3270,3270,3270,3270,3270,3270,3270,3270,3270,3270,  
 3270,3270,3270,3270,3269,3269,3269,3269,3269,3269,3269,3269,3269,3269,3269,3269,3269,  
 3269,3269,3269,3269,3269,3269,3269,3269,3268,3268,3268,3268,3268,3268,3268,3268,3268,  
 3268,3268,3268,3268,3268,3268,3267,3267,3267,3267,3267,3267,3267,3267,3267,3267,3267,  
 3267,3267,3267,3267,3267,3267,3266,3266,3266,3266,3266,3266,3266,3266,3266,3266,3266,  
 3266,3266,3265,3265,3265,3265,3265,3265,3265,3265,3265,3265,3264,3264,3264,3264,3264,  
 3264,3264,3264,3264,3264,3263,3263,3263,3263,3263,3263,3262,3262,3262,3262,3262,3262,  
 3262,3262,3261,3261,3261,3261,3261,3261,3260,3260,3260,3260,3259,3259,3259,3259,3259,  
 3258,3258,3258,3258,3258,3257,3257,3257,3257,3257,3256,3256,3255,3255,3255,3255,3254,  
 3254,3254,3253,3253,3253,3253,3252,3252,3252,3252,3251,3251,3251,3250,3250,3250,3249,3249,  
 3249,3249,3248,3248,3248,3248,3247,3247,3247,3247,3247,3246,3246,3246,3246,3245,3245,3245,  
 3244,3244,3244,3244,3244,3243,3243,3243,3243,3243,3242,3242,3242,3242,3242,3242,3241,  
 3241,3241,3241,3240,3240,3240,3240,3240,3240,3239,3239,3239,3239,3239,3238,3238,3238,3238,  
 3238,3238,3237,3237,3237,3237,3237,3236,3236,3236,3236,3236,3236,3235,3235,3235,3235,3235,  
 3235,3235,3234,3234,3234,3234,3234,3233,3233,3233,3233,3233,3233,3233,3232,3232,3232,3232,  
 3232,3232,3232,3232,3231,3231,3231,3231,3231,3231,3230,3230,3230,3230,3230,3230,3230,3230,  
 3229,3229,3229,3229,3229,3229,3229,3229,3228,3228,3228,3228,3228,3228,3227,3227,3227,3227,  
 3227,3227,3227,3227,3227,3226,3226,3226,3226,3226,3226,3225,3225,3225,3225,3225,3225,  
 3225,3224,3224,3224,3224,3224,3224,3224,3223,3223,3223,3223,3223,3222,3222,3222,3222,  
 3222,3221,3221,3221,3221,3221,3221,3220,3220,3220,3220,3219,3219,3219,3219,3218,3218,3218,  
 3218,3218,3217,3217,3217,3217,3217,3216,3216,3216,3216,3215,3215,3215,3215,3214,3214,3214,  
 3213,3213,3213,3213,3212,3212,3212,3212,3211,3211,3211,3211,3210,3210,3210,3210,3209,3209,  
 3209,3209,3208,3208,3208,3207,3207,3207,3207,3206,3206,3206,3206,3205,3205,3205,3205,3204,  
 3204,3204,3204,3203,3203,3203,3202,3202,3202,3202,3201,3201,3201,3201,3200,3200,3200,3200,  
 3200,3199,3199,3199,3198,3198,3198,3197,3197,3197,3197,3196,3196,3196,3195,3195,3195,3195,  
 3194,3194,3194,3193,3193,3193,3193,3192,3192,3192,3191,3191,3190,3190,3190,3189,3189,3189,  
 3188,3188,3187,3187,3187,3186,3186,3186,3185,3185,3184,3184,3184,3183,3183,3182,3182,3181,  
 3181,3181,3180,3180,3180,3179,3179,3179,3178,3178,3178,3178,3177,3177,3177,3177,3176,3176,

3176,3176,3176,3175,3175,3175,3175,3175,3175,3175,3175,3174,3174,3174,3174,3174,3174,3174,3173,3173,3173,3173,3173,3173,3173,3173,3172,3172,3172,3172,3172,3172,3172,3172,3172,3171,3171,3171,3171,3171,3170,3170,3170,3170,3170,3170,3170,3169,3169,3169,3169,3169,3169,3168,3168,3168,3168,3167,3167,3167,3167,3167,3166,3166,3166,3166,3166,3166,3165,3165,3165,3164,3164,3164,3164,3163,3163,3162,3162,3162,3161,3161,3160,3160,3160,3159,3159,3158,3158,3157,3156,3156,3155,3154,3153,3152,3151,3150,3149,3148,3147,3145,3144,3143,3141,3140,3138,3137,3135,3133,3132,3130,3128,3126,3124,3122,3120,3118,3116,3114,3112,3110,3108,3106,3104,3102,3100,3097,3095,3093,3091,3088,3086,3084,3081,3079,3076,3074,3071,3069,3066,3064,3061,3058,3056,3053,3050,3047,3044,3042,3039,3036,3033,3030,3027,3024,3021,3017,3014,3011,3007,3004,3001,2997,2994,2990,2986,2983,2979,2975,2971,2967,2963,2959,2955,2951,2947,2942,2938,2933,2929,2924,2919,2914,2909,2904,2899,2893,2888,2882,2877,2871,2865,2858,2852,2845,2839,2832,2824,2817,2809,2801,2793,2784,2776,2766,2757,2747,2736,2725,2714,2701,2688,2674,2658,2640,2620,2599,2575,2549,2518,2500};

const int32

soc\_table\_20[871]={5793,5787,5780,5773,5767,5760,5753,5747,5740,5733,5727,5720,5713,5707,5700,5694,5687,5680,5674,5667,5660,5654,5647,5640,5634,5627,5620,5614,5607,5600,5594,5587,5580,5574,5567,5560,5554,5547,5540,5534,5527,5521,5514,5507,5501,5494,5487,5481,5474,5467,5461,5454,5447,5441,5434,5427,5421,5414,5407,5401,5394,5387,5381,5374,5367,5361,5354,5348,5341,5334,5328,5321,5314,5308,5301,5294,5288,5281,5274,5268,5261,5254,5248,5241,5234,5228,5221,5214,5208,5201,5194,5188,5181,5174,5168,5161,5155,5148,5141,5135,5128,5121,5115,5108,5101,5095,5088,5081,5075,5068,5061,5055,5048,5041,5035,5028,5021,5015,5008,5001,4995,4988,4982,4975,4968,4962,4955,4948,4942,4935,4928,4922,4915,4908,4902,4895,4888,4882,4875,4868,4862,4855,4848,4842,4835,4828,4822,4815,4808,4802,4795,4789,4782,4775,4769,4762,4755,4749,4742,4735,4729,4722,4715,4709,4702,4695,4689,4682,4675,4669,4662,4655,4649,4642,4635,4629,4622,4616,4609,4602,4596,4589,4582,4576,4569,4562,4556,4549,4542,4536,4529,4522,4516,4509,4502,4496,4489,4482,4476,4469,4462,4456,4449,4443,4436,4429,4423,4416,4409,4403,4396,4389,4383,4376,4369,4363,4356,4349,4343,4336,4329,4323,4316,4309,4303,4296,4289,4283,4276,4269,4263,4256,4250,4243,4236,4230,4223,4216,4210,4203,4196,4190,4183,4176,4170,4163,4156,4150,4143,4136,4130,4123,4116,4110,4103,4096,4090,4083,4077,4070,4063,4057,4050,4043,4037,4030,4023,4017,4010,4003,3997,3990,3983,3977,3970,3963,3957,3950,3943,3937,3930,3923,3917,3910,3903,3897,3890,3884,3877,3870,3864,3857,3850,3844,3837,3830,3824,3817,3810,3804,3797,3790,3784,3777,3770,3764,3757,3750,3744,3737,3730,3724,3717,3711,3704,3697,3691,3684,3677,3671,3664,3657,3651,3644,3637,3631,3624,3617,3611,3604,3597,3591,3584,3577,3571,3564,3557,3551,3544,3538,3531,3524,3518,3511,3504,3498,3491,3484,3478,3471,3464,3458,3451,3444,3438,3431,3424,3418,3411,3404,3398,3391,3384,3378,3371,3364,3358,3351,3345,3338,3331,3325,3318,3311,3305,3298,3291,3285,3278,3271,3265,3258,3251,3245,3238,3231,3225,3218,3211,3205,3198,3191,3185,3178,3172,3165,3158,3152,3145,3138,3132,3125,3118,3112,3105,3098,3092,3085,3078,3072,3065,3058,3052,3045,3038,3032,3025,3018,3012,3005,2998,2992,2985,2978,2972,2965,2958,2952,2945,2938,2932,2925,2918,2912,2905,2898,2892,2885,2878,2872,2865,2858,2852,2845,2838,2832,2825,2818,2812,2805,2798,2792,2785,2778,2772,2765,2758,2752,2745,2738,2732,2725,2718,2712,2705,2698,2692,2685,2678,2672,2665,2658,2652,2645,2638,2632,2625,2618,2612,2605,2598,2592,2585,2578,2572,2565,2558,2552,2545,2538,2532,2525,2518,2512,2505,2498,2492,2485,2478,2472,2465,2458,2452,2445,2438,2432,2425,2418,2412,2405,2398,2392,2385,2378,2372,2365,2358,2352,2345,2338,2332,2325,2318,2312,2305,2298,2292,2285,2278,2272,2265,2258,2252,2245,2238,2232,2225,2218,2212,2205,2198,2192,2185,2178,2172,2165,2158,2152,2145,2138,2132,2125,2118,2112,2105,2098,2092,2085,2078,2072,2065,2058,2052,2045,2038,2032,2025,2018,2012,2005,1998,1992,1985,1978,1972,1965,1958,1952,1945,1938,1932,1925,1918,1912,1905,1898,1892,1885,1878,1872,1865,1858,1852,1845,1838,1832,1825,1818,1812,1805,1798,1792,1785,1778,1772,1765,1758,1752,1745,1738,1732,1725,1718,1712,1705,1698,1692,1685,1678,1672,1665,1658,1652,1645,1638,1632,1625,1618,1612,1605,1598,1592,1585,1578,1572,1565,1558,1552,1545,1538,1532,1525,1518,1512,1505,1498,1492,1485,1478,1472,1465,1458,1452,1445,1438,1432,1425,1418,1412,1405,1398,1392,1385,1378,1372,1365,1358,1352,1345,1338,1332,1325,1318,1312,1305,1298,1292,1285,1278,1272,1265,1258,1252,1245,1238,1232,1225,1218,1212,1205,1198,1192,1185,1178,1172,1165,1158,1152,1145,1138,1132,1125,1118,1112,1105,1098,1092,1085,1078,1072,1065,1058,1052,1045,1038,1032,1025,1018,1012,1005,998,992,985,978,972,965,958,952,945,938,932,925,918,912,905,898,892,885,878,872,865,858,852,845,838,832,825,818,812,805,798,792,785,778,772,765,758,752,745,738,732,725,718,712,705,698,692,685,678,672,665,658,652,645,638,632,625,618,612,605,598,592,585,578,572,565,558,552,545,538,532,525,518,512,505,498,492,485,478,472,465,458,452,445,438,432,425,418,412,405,398,392,385,378,372,365,358,352,345,338,332,325,318,312,305,298,292,285,278,272,265,258,252,245,238,232,225,218,212,205,198,192,185,178,172,165,158,152,145,138,132,125,118,112,105,98,92,85,78,72,65,58,52,45,38,32,25,18,12,5,0};

065,3058,3052,3045,3038,3032,3025,3018,3012,3005,2998,2992,2985,2979,2972,2965,2959,2952,2  
 945,2939,2932,2925,2919,2912,2905,2899,2892,2885,2879,2872,2865,2859,2852,2845,2839,2832,2  
 825,2819,2812,2806,2799,2792,2786,2779,2772,2766,2759,2752,2746,2739,2732,2726,2719,2712,2  
 706,2699,2692,2686,2679,2672,2666,2659,2652,2646,2639,2633,2626,2619,2613,2606,2599,2593,2  
 586,2579,2573,2566,2559,2553,2546,2539,2533,2526,2519,2513,2506,2499,2493,2486,2479,2473,2  
 466,2459,2453,2446,2440,2433,2426,2420,2413,2406,2400,2393,2386,2380,2373,2366,2360,2353,2  
 346,2340,2333,2326,2320,2313,2306,2300,2293,2286,2280,2273,2267,2260,2253,2247,2240,2233,2  
 227,2220,2213,2207,2200,2193,2187,2180,2173,2167,2160,2153,2147,2140,2133,2127,2120,2113,2  
 107,2100,2093,2087,2080,2074,2067,2060,2054,2047,2040,2034,2027,2020,2014,2007,2000,1994,1  
 987,1980,1974,1967,1960,1954,1947,1940,1934,1927,1920,1914,1907,1901,1894,1887,1881,1874,1  
 867,1861,1854,1847,1841,1834,1827,1821,1814,1807,1801,1794,1787,1781,1774,1767,1761,1754,1  
 747,1741,1734,1728,1721,1714,1708,1701,1694,1688,1681,1674,1668,1661,1654,1648,1641,1634,1  
 628,1621,1614,1608,1601,1594,1588,1581,1574,1568,1561,1554,1548,1541,1535,1528,1521,1515,1  
 508,1501,1495,1488,1481,1475,1468,1461,1455,1448,1441,1435,1428,1421,1415,1408,1401,1395,1  
 388,1381,1375,1368,1362,1355,1348,1342,1335,1328,1322,1315,1308,1302,1295,1288,1282,1275,1  
 268,1262,1255,1248,1242,1235,1228,1222,1215,1208,1202,1195,1189,1182,1175,1169,1162,1155,1  
 149,1142,1135,1129,1122,1115,1109,1102,1095,1089,1082,1075,1069,1062,1055,1049,1042,1035,1  
 029,1022,1015,1009,1002,996,989,982,976,969,962,956,949,942,936,929,922,916,909,902,896,889,  
 882,876,869,862,856,849,842,836,829,823,816,809,803,796,789,783,776,769,763,756,749,743,736,7  
 29,723,716,709,703,696,689,683,676,669,663,656,650,643,636,630,623,616,610,603,596,590,583,57  
 6,570,563,556,550,543,536,530,523,516,510,503,496,490,483,476,470,463,457,450,443,437,430,423  
 ,417,410,403,397,390,383,377,370,363,357,350,343,337,330,323,317,310,303,297,290,284,277,270,  
 264,257,250,244,237,230,224,217,210,204,197,190,184,177,170,164,157,150,144,137,130,124,117,1  
 11,104,97,91,84,77,71,64,57,51,44,37,31,24,17,11,4};

const

int16

V\_soc\_40[1544]={3531,3504,3486,3472,3461,3453,3446,3440,3435,3431,3428,3425,3423,3421,341  
 9,3418,3417,3416,3415,3414,3414,3413,3413,3412,3412,3412,3412,3412,3411,3406,3396,3387,338  
 1,3376,3373,3370,3367,3365,3363,3362,3361,3360,3359,3358,3357,3357,3356,3355,3355,3354,335  
 4,3354,3353,3353,3352,3352,3352,3351,3351,3351,3351,3350,3350,3350,3350,3350,3349,3349,334  
 9,3349,3349,3349,3348,3348,3348,3348,3348,3348,3348,3347,3347,3347,3347,3347,3347,3347,334  
 7,3347,3347,3346,3346,3346,3346,3346,3346,3346,3346,3346,3346,3346,3346,3346,3345,3345,334  
 5,3345,3345,3345,3345,3345,3345,3345,3345,3345,3345,3345,3345,3345,3345,3344,3344,3344,334  
 4,3344,3344,3344,3344,3344,3344,3344,3344,3344,3344,3344,3344,3344,3344,3344,3344,3344,334  
 4,3344,3344,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,334  
 3,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3343,3342,3342,3342,3342,334  
 2,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,334  
 2,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3342,3341,3341,3341,3341,3341,334





```

const int32
soc_table_40[1544]={10000,9994,9987,9981,9974,9968,9961,9955,9948,9942,9935,9929,9922,9916
,9909,9903,9896,9890,9883,9877,9871,9864,9858,9851,9845,9838,9832,9825,9819,9812,9806,9799
,9793,9786,9780,9773,9767,9761,9754,9748,9741,9735,9728,9722,9715,9709,9702,9696,9689,9683
,9676,9670,9663,9657,9650,9644,9638,9631,9625,9618,9612,9605,9599,9592,9586,9579,9573,9566
,9560,9553,9547,9540,9534,9528,9521,9515,9508,9502,9495,9489,9482,9476,9469,9463,9456,9450
,9443,9437,9430,9424,9417,9411,9405,9398,9392,9385,9379,9372,9366,9359,9353,9346,9340,9333
,9327,9320,9314,9307,9301,9294,9288,9282,9275,9269,9262,9256,9249,9243,9236,9230,9223,9217
,9210,9204,9197,9191,9184,9178,9172,9165,9159,9152,9146,9139,9133,9126,9120,9113,9107,9100
,9094,9087,9081,9074,9068,9061,9055,9049,9042,9036,9029,9023,9016,9010,9003,8997,8990,8984
,8977,8971,8964,8958,8951,8945,8939,8932,8926,8919,8913,8906,8900,8893,8887,8880,8874,8867
,8861,8854,8848,8841,8835,8828,8822,8816,8809,8803,8796,8790,8783,8777,8770,8764,8757,8751
,8744,8738,8731,8725,8718,8712,8706,8699,8693,8686,8680,8673,8667,8660,8654,8647,8641,8634
,8628,8621,8615,8608,8602,8595,8589,8583,8576,8570,8563,8557,8550,8544,8537,8531,8524,8518
,8511,8505,8498,8492,8485,8479,8472,8466,8460,8453,8447,8440,8434,8427,8421,8414,8408,8401
,8395,8388,8382,8375,8369,8362,8356,8350,8343,8337,8330,8324,8317,8311,8304,8298,8291,8285
,8278,8272,8265,8259,8252,8246,8239,8233,8227,8220,8214,8207,8201,8194,8188,8181,8175,8168
,8162,8155,8149,8142,8136,8129,8123,8117,8110,8104,8097,8091,8084,8078,8071,8065,8058,8052
,8045,8039,8032,8026,8019,8013,8006,8000,7994,7987,7981,7974,7968,7961,7955,7948,7942,7935
,7929,7922,7916,7909,7903,7896,7890,7883,7877,7871,7864,7858,7851,7845,7838,7832,7825,7819
,7812,7806,7799,7793,7786,7780,7773,7767,7761,7754,7748,7741,7735,7728,7722,7715,7709,7702
,7696,7689,7683,7676,7670,7663,7657,7650,7644,7638,7631,7625,7618,7612,7605,7599,7592,7586
,7579,7573,7566,7560,7553,7547,7540,7534,7528,7521,7515,7508,7502,7495,7489,7482,7476,7469
,7463,7456,7450,7443,7437,7430,7424,7417,7411,7405,7398,7392,7385,7379,7372,7366,7359,7353
,7346,7340,7333,7327,7320,7314,7307,7301,7294,7288,7282,7275,7269,7262,7256,7249,7243,7236
,7230,7223,7217,7210,7204,7197,7191,7184,7178,7172,7165,7159,7152,7146,7139,7133,7126,7120
,7113,7107,7100,7094,7087,7081,7074,7068,7061,7055,7049,7042,7036,7029,7023,7016,7010,7003
,6997,6990,6984,6977,6971,6964,6958,6951,6945,6939,6932,6926,6919,6913,6906,6900,6893,6887
,6880,6874,6867,6861,6854,6848,6841,6835,6828,6822,6816,6809,6803,6796,6790,6783,6777,6770
,6764,6757,6751,6744,6738,6731,6725,6718,6712,6706,6699,6693,6686,6680,6673,6667,6660,6654
,6647,6641,6634,6628,6621,6615,6608,6602,6595,6589,6583,6576,6570,6563,6557,6550,6544,6537
,6531,6524,6518,6511,6505,6498,6492,6485,6479,6472,6466,6460,6453,6447,6440,6434,6427,6421
,6414,6408,6401,6395,6388,6382,6375,6369,6362,6356,6350,6343,6337,6330,6324,6317,6311,6304
,6298,6291,6285,6278,6272,6265,6259,6252,6246,6239,6233,6227,6220,6214,6207,6201,6194,6188
,6181,6175,6168,6162,6155,6149,6142,6136,6129,6123,6117,6110,6104,6097,6091,6084,6078,6071
,6065,6058,6052,6045,6039,6032,6026,6019,6013,6006,6000,5994,5987,5981,5974,5968,5961,5955
,5948,5942,5935,5929,5922,5916,5909,5903,5896,5890,5883,5877,5871,5864,5858,5851,5845,5838

```

,5832,5825,5819,5812,5806,5799,5793,5786,5780,5773,5767,5761,5754,5748,5741,5735,5728,5722  
,5715,5709,5702,5696,5689,5683,5676,5670,5663,5657,5650,5644,5638,5631,5625,5618,5612,5605  
,5599,5592,5586,5579,5573,5566,5560,5553,5547,5540,5534,5528,5521,5515,5508,5502,5495,5489  
,5482,5476,5469,5463,5456,5450,5443,5437,5430,5424,5417,5411,5405,5398,5392,5385,5379,5372  
,5366,5359,5353,5346,5340,5333,5327,5320,5314,5307,5301,5294,5288,5282,5275,5269,5262,5256  
,5249,5243,5236,5230,5223,5217,5210,5204,5197,5191,5184,5178,5172,5165,5159,5152,5146,5139  
,5133,5126,5120,5113,5107,5100,5094,5087,5081,5074,5068,5061,5055,5049,5042,5036,5029,5023  
,5016,5010,5003,4997,4990,4984,4977,4971,4964,4958,4951,4945,4939,4932,4926,4919,4913,4906  
,4900,4893,4887,4880,4874,4867,4861,4854,4848,4841,4835,4828,4822,4816,4809,4803,4796,4790  
,4783,4777,4770,4764,4757,4751,4744,4738,4731,4725,4718,4712,4706,4699,4693,4686,4680,4673  
,4667,4660,4654,4647,4641,4634,4628,4621,4615,4608,4602,4595,4589,4583,4576,4570,4563,4557  
,4550,4544,4537,4531,4524,4518,4511,4505,4498,4492,4485,4479,4472,4466,4460,4453,4447,4440  
,4434,4427,4421,4414,4408,4401,4395,4388,4382,4375,4369,4362,4356,4350,4343,4337,4330,4324  
,4317,4311,4304,4298,4291,4285,4278,4272,4265,4259,4252,4246,4239,4233,4227,4220,4214,4207  
,4201,4194,4188,4181,4175,4168,4162,4155,4149,4142,4136,4129,4123,4117,4110,4104,4097,4091  
,4084,4078,4071,4065,4058,4052,4045,4039,4032,4026,4019,4013,4006,4000,3994,3987,3981,3974  
,3968,3961,3955,3948,3942,3935,3929,3922,3916,3909,3903,3896,3890,3883,3877,3871,3864,3858  
,3851,3845,3838,3832,3825,3819,3812,3806,3799,3793,3786,3780,3773,3767,3761,3754,3748,3741  
,3735,3728,3722,3715,3709,3702,3696,3689,3683,3676,3670,3663,3657,3650,3644,3638,3631,3625  
,3618,3612,3605,3599,3592,3586,3579,3573,3566,3560,3553,3547,3540,3534,3528,3521,3515,3508  
,3502,3495,3489,3482,3476,3469,3463,3456,3450,3443,3437,3430,3424,3417,3411,3405,3398,3392  
,3385,3379,3372,3366,3359,3353,3346,3340,3333,3327,3320,3314,3307,3301,3294,3288,3282,3275  
,3269,3262,3256,3249,3243,3236,3230,3223,3217,3210,3204,3197,3191,3184,3178,3172,3165,3159  
,3152,3146,3139,3133,3126,3120,3113,3107,3100,3094,3087,3081,3074,3068,3061,3055,3049,3042  
,3036,3029,3023,3016,3010,3003,2997,2990,2984,2977,2971,2964,2958,2951,2945,2939,2932,2926  
,2919,2913,2906,2900,2893,2887,2880,2874,2867,2861,2854,2848,2841,2835,2828,2822,2816,2809  
,2803,2796,2790,2783,2777,2770,2764,2757,2751,2744,2738,2731,2725,2718,2712,2706,2699,2693  
,2686,2680,2673,2667,2660,2654,2647,2641,2634,2628,2621,2615,2608,2602,2595,2589,2583,2576  
,2570,2563,2557,2550,2544,2537,2531,2524,2518,2511,2505,2498,2492,2485,2479,2472,2466,2460  
,2453,2447,2440,2434,2427,2421,2414,2408,2401,2395,2388,2382,2375,2369,2362,2356,2350,2343  
,2337,2330,2324,2317,2311,2304,2298,2291,2285,2278,2272,2265,2259,2252,2246,2239,2233,2227  
,2220,2214,2207,2201,2194,2188,2181,2175,2168,2162,2155,2149,2142,2136,2129,2123,2117,2110  
,2104,2097,2091,2084,2078,2071,2065,2058,2052,2045,2039,2032,2026,2019,2013,2006,2000,1994  
,1987,1981,1974,1968,1961,1955,1948,1942,1935,1929,1922,1916,1909,1903,1896,1890,1883,1877  
,1871,1864,1858,1851,1845,1838,1832,1825,1819,1812,1806,1799,1793,1786,1780,1773,1767,1761  
,1754,1748,1741,1735,1728,1722,1715,1709,1702,1696,1689,1683,1676,1670,1663,1657,1650,1644  
,1638,1631,1625,1618,1612,1605,1599,1592,1586,1579,1573,1566,1560,1553,1547,1540,1534,1528







0,3270,3270,3270,3269,3269,3269,3269,3269,3269,3269,3268,3268,3268,3268,3268,3268,3267,3267,3267,3267,3267,3266,3266,3266,3266,3266,3266,3265,3265,3265,3265,3265,3264,3264,3264,3264,3263,3263,3263,3263,3262,3262,3262,3262,3262,3261,3261,3261,3261,3260,3260,3260,3260,3259,3259,3259,3259,3259,3258,3258,3258,3258,3257,3257,3257,3257,3256,3256,3256,3256,3255,3255,3255,3255,3255,3254,3254,3254,3254,3253,3253,3253,3253,3252,3252,3252,3252,3252,3252,3251,3251,3250,3250,3250,3250,3249,3249,3249,3249,3249,3248,3248,3248,3247,3247,3247,3247,3246,3246,3246,3246,3246,3245,3245,3245,3244,3244,3244,3244,3244,3243,3243,3243,3243,3242,3242,3241,3241,3241,3240,3240,3240,3240,3240,3239,3239,3239,3238,3238,3237,3237,3237,3237,3236,3236,3236,3236,3235,3235,3234,3234,3234,3233,3233,3233,3232,3232,3232,3232,3231,3231,3230,3230,3230,3229,3229,3228,3228,3228,3228,3227,3227,3227,3226,3226,3226,3225,3225,3225,3224,3224,3223,3223,3223,3223,3223,3222,3222,3222,3221,3221,3221,3220,3220,3220,3220,3219,3219,3219,3219,3218,3218,3218,3218,3217,3217,3217,3217,3217,3217,3217,3216,3216,3216,3215,3215,3215,3215,3215,3215,3215,3215,3214,3214,3214,3214,3213,3213,3213,3213,3213,3213,3212,3212,3212,3212,3211,3211,3211,3211,3211,3211,3211,3211,3210,3210,3210,3209,3209,3209,3209,3209,3209,3209,3209,3208,3208,3208,3207,3207,3207,3207,3206,3206,3206,3206,3205,3205,3205,3204,3204,3204,3204,3203,3203,3203,3202,3202,3201,3201,3201,3200,3200,3199,3199,3198,3197,3197,3196,3195,3195,3194,3193,3192,3192,3192,3191,3190,3189,3188,3187,3186,3185,3184,3183,3182,3181,3180,3179,3178,3177,3175,3174,3173,3173,3172,3170,3169,3167,3166,3164,3163,3161,3159,3158,3156,3154,3152,3150,3149,3147,3145,3143,3143,3142,3139,3137,3135,3133,3131,3129,3127,3125,3123,3121,3119,3116,3114,3112,3109,3107,3104,3104,3103,3100,3097,3095,3092,3090,3087,3085,3082,3079,3076,3073,3070,3068,3065,3062,3059,3056,3056,3055,3050,3047,3043,3040,3036,3033,3030,3026,3023,3019,3016,3012,3008,3004,3000,2996,2992,2988,2984,2979,2975,2971,2966,2961,2956,2951,2946,2941,2935,2930,2924,2918,2913,2906,2900,2896,2892,2886,2879,2872,2864,2856,2848,2839,2830,2820,2808,2796,2783,2769,2754,2738,2726};

const int32 soc\_table\_60[1544]={10000,9994,9987,9981,9975,9968,9962,9956,9949,9943,9936,9930,9924,9917,9911,9905,9898,9892,9886,9879,9873,9867,9860,9854,9848,9841,9835,9829,9822,9816,9809,9803,9797,9790,9784,9778,9771,9765,9759,9752,9746,9740,9733,9727,9721,9714,9708,9701,9695,9689,9682,9676,9670,9663,9657,9651,9644,9638,9632,9625,9619,9613,9606,9600,9594,9587,9581,9574,9568,9562,9555,9549,9543,9536,9530,9524,9517,9511,9505,9498,9492,9486,9479,9473,9466,9460,9454,9447,9441,9435,9428,9422,9416,9409,9403,9397,9390,9384,9378,9371,9365,9359,9352,9346,9339,9333,9327,9320,9314,9308,9301,9295,9289,9282,9276,9270,9263,9257,9251,9244,9238,9231,9225,9219,9212,9206,9200,9193,9187,9181,9174,9168,9162,9155,9149,9143,9136,9130,9124,9117,9111,9104,9098,9092,9085,9079,9073,9066,9060,9054,9047,9041,9035,9028,9022,9016,9009,9003,8997,8990,8984,8977,8971,8965,8958,8952,8946,8939,8933,8927,8920,8914,8908,8901,8895,8889,8882,8876,8869,8863,8857,8850,8844,8838,8831,8825,8819,8812,8806,8800,8793,8787,8781,8774,8768,8762,8755,8749,8742,8736,8730,8723,8717,8711,8704,8698,8692,8685,8679,8673,8666,8660

,8654,8647,8641,8634,8628,8622,8615,8609,8603,8596,8590,8584,8577,8571,8565,8558,8552,8546  
,8539,8533,8527,8520,8514,8507,8501,8495,8488,8482,8476,8469,8463,8457,8450,8444,8438,8431  
,8425,8419,8412,8406,8399,8393,8387,8380,8374,8368,8361,8355,8349,8342,8336,8330,8323,8317  
,8311,8304,8298,8292,8285,8279,8272,8266,8260,8253,8247,8241,8234,8228,8222,8215,8209,8203  
,8196,8190,8184,8177,8171,8164,8158,8152,8145,8139,8133,8126,8120,8114,8107,8101,8095,8088  
,8082,8076,8069,8063,8057,8050,8044,8037,8031,8025,8018,8012,8006,7999,7993,7987,7980,7974  
,7968,7961,7955,7949,7942,7936,7929,7923,7917,7910,7904,7898,7891,7885,7879,7872,7866,7860  
,7853,7847,7841,7834,7828,7822,7815,7809,7802,7796,7790,7783,7777,7771,7764,7758,7752,7745  
,7739,7733,7726,7720,7714,7707,7701,7694,7688,7682,7675,7669,7663,7656,7650,7644,7637,7631  
,7625,7618,7612,7606,7599,7593,7587,7580,7574,7567,7561,7555,7548,7542,7536,7529,7523,7517  
,7510,7504,7498,7491,7485,7479,7472,7466,7459,7453,7447,7440,7434,7428,7421,7415,7409,7402  
,7396,7390,7383,7377,7371,7364,7358,7352,7345,7339,7332,7326,7320,7313,7307,7301,7294,7288  
,7282,7275,7269,7263,7256,7250,7244,7237,7231,7224,7218,7212,7205,7199,7193,7186,7180,7174  
,7167,7161,7155,7148,7142,7136,7129,7123,7117,7110,7104,7097,7091,7085,7078,7072,7066,7059  
,7053,7047,7040,7034,7028,7021,7015,7009,7002,6996,6990,6983,6977,6970,6964,6958,6951,6945  
,6939,6932,6926,6920,6913,6907,6901,6894,6888,6882,6875,6869,6862,6856,6850,6843,6837,6831  
,6824,6818,6812,6805,6799,6793,6786,6780,6774,6767,6761,6755,6748,6742,6735,6729,6723,6716  
,6710,6704,6697,6691,6685,6678,6672,6666,6659,6653,6647,6640,6634,6627,6621,6615,6608,6602  
,6596,6589,6583,6577,6570,6564,6558,6551,6545,6539,6532,6526,6520,6513,6507,6500,6494,6488  
,6481,6475,6469,6462,6456,6450,6443,6437,6431,6424,6418,6412,6405,6399,6392,6386,6380,6373  
,6367,6361,6354,6348,6342,6335,6329,6323,6316,6310,6304,6297,6291,6285,6278,6272,6265,6259  
,6253,6246,6240,6234,6227,6221,6215,6208,6202,6196,6189,6183,6177,6170,6164,6157,6151,6145  
,6138,6132,6126,6119,6113,6107,6100,6094,6088,6081,6075,6069,6062,6056,6050,6043,6037,6030  
,6024,6018,6011,6005,5999,5992,5986,5980,5973,5967,5961,5954,5948,5942,5935,5929,5922,5916  
,5910,5903,5897,5891,5884,5878,5872,5865,5859,5853,5846,5840,5834,5827,5821,5815,5808,5802  
,5795,5789,5783,5776,5770,5764,5757,5751,5745,5738,5732,5726,5719,5713,5707,5700,5694,5687  
,5681,5675,5668,5662,5656,5649,5643,5637,5630,5624,5618,5611,5605,5599,5592,5586,5580,5573  
,5567,5560,5554,5548,5541,5535,5529,5522,5516,5510,5503,5497,5491,5484,5478,5472,5465,5459  
,5452,5446,5440,5433,5427,5421,5414,5408,5402,5395,5389,5383,5376,5370,5364,5357,5351,5345  
,5338,5332,5325,5319,5313,5306,5300,5294,5287,5281,5275,5268,5262,5256,5249,5243,5237,5230  
,5224,5217,5211,5205,5198,5192,5186,5179,5173,5167,5160,5154,5148,5141,5135,5129,5122,5116  
,5110,5103,5097,5090,5084,5078,5071,5065,5059,5052,5046,5040,5033,5027,5021,5014,5008,5002  
,4995,4989,4983,4976,4970,4963,4957,4951,4944,4938,4932,4925,4919,4913,4906,4900,4894,4887  
,4881,4875,4868,4862,4855,4849,4843,4836,4830,4824,4817,4811,4805,4798,4792,4786,4779,4773  
,4767,4760,4754,4748,4741,4735,4728,4722,4716,4709,4703,4697,4690,4684,4678,4671,4665,4659  
,4652,4646,4640,4633,4627,4620,4614,4608,4601,4595,4589,4582,4576,4570,4563,4557,4551,4544  
,4538,4532,4525,4519,4513,4506,4500,4493,4487,4481,4474,4468,4462,4455,4449,4443,4436,4430

,4424,4417,4411,4405,4398,4392,4385,4379,4373,4366,4360,4354,4347,4341,4335,4328,4322,4316,4309,4303,4297,4290,4284,4278,4271,4265,4258,4252,4246,4239,4233,4227,4220,4214,4208,4201,4195,4189,4182,4176,4170,4163,4157,4150,4144,4138,4131,4125,4119,4112,4106,4100,4093,4087,4081,4074,4068,4062,4055,4049,4043,4036,4030,4023,4017,4011,4004,3998,3992,3985,3979,3973,3966,3960,3954,3947,3941,3935,3928,3922,3915,3909,3903,3896,3890,3884,3877,3871,3865,3858,3852,3846,3839,3833,3827,3820,3814,3808,3801,3795,3788,3782,3776,3769,3763,3757,3750,3744,3738,3731,3725,3719,3712,3706,3700,3693,3687,3680,3674,3668,3661,3655,3649,3642,3636,3630,3623,3617,3611,3604,3598,3592,3585,3579,3573,3566,3560,3553,3547,3541,3534,3528,3522,3515,3509,3503,3496,3490,3484,3477,3471,3465,3458,3452,3445,3439,3433,3426,3420,3414,3407,3401,3395,3388,3382,3376,3369,3363,3357,3350,3344,3338,3331,3325,3318,3312,3306,3299,3293,3287,3280,3274,3268,3261,3255,3249,3242,3236,3230,3223,3217,3210,3204,3198,3191,3185,3179,3172,3166,3160,3153,3147,3141,3134,3128,3122,3115,3109,3103,3096,3090,3083,3077,3071,3064,3058,3052,3045,3039,3033,3026,3020,3014,3007,3001,2995,2988,2982,2976,2969,2963,2956,2950,2944,2937,2931,2925,2918,2912,2906,2899,2893,2887,2880,2874,2868,2861,2855,2848,2842,2836,2829,2823,2817,2810,2804,2798,2791,2785,2779,2772,2766,2760,2753,2747,2741,2734,2728,2721,2715,2709,2702,2696,2690,2683,2677,2671,2664,2658,2652,2645,2639,2633,2626,2620,2613,2607,2601,2594,2588,2582,2575,2569,2563,2556,2550,2544,2537,2531,2525,2518,2512,2506,2499,2493,2486,2480,2474,2467,2461,2455,2448,2442,2436,2429,2423,2417,2410,2404,2398,2391,2385,2378,2372,2366,2359,2353,2347,2340,2334,2328,2321,2315,2309,2302,2296,2290,2283,2277,2271,2264,2258,2251,2245,2239,2232,2226,2220,2213,2207,2201,2194,2188,2182,2175,2169,2163,2156,2150,2143,2137,2131,2124,2118,2112,2105,2099,2093,2086,2080,2074,2067,2061,2055,2048,2042,2036,2029,2023,2016,2010,2004,1997,1991,1985,1978,1972,1966,1959,1953,1947,1940,1934,1928,1921,1915,1908,1902,1896,1889,1883,1877,1870,1864,1858,1851,1845,1839,1832,1826,1820,1813,1807,1801,1794,1788,1781,1775,1769,1762,1756,1750,1743,1737,1731,1724,1718,1712,1705,1699,1693,1686,1680,1673,1667,1661,1654,1648,1642,1635,1629,1623,1616,1610,1604,1597,1591,1585,1578,1572,1566,1559,1553,1546,1540,1534,1527,1521,1515,1508,1502,1496,1489,1483,1477,1470,1464,1458,1451,1445,1438,1432,1426,1419,1413,1407,1400,1394,1388,1381,1375,1369,1362,1356,1350,1343,1337,1331,1324,1318,1311,1305,1299,1292,1286,1280,1273,1267,1261,1254,1248,1242,1235,1229,1223,1216,1210,1203,1197,1191,1184,1178,1172,1165,1159,1153,1146,1140,1134,1127,1121,1115,1108,1102,1096,1089,1083,1076,1070,1064,1057,1051,1045,1038,1032,1026,1019,1013,1007,1000,994,988,981,975,969,962,956,949,943,937,930,924,918,911,905,899,892,886,880,873,867,861,854,848,841,835,829,822,816,810,803,797,791,784,778,772,765,759,753,746,740,734,727,721,714,708,702,695,689,683,676,670,664,657,651,645,638,632,626,619,613,606,600,594,587,581,575,568,562,556,549,543,537,530,524,518,511,505,499,492,486,479,473,467,460,454,448,441,435,429,422,416,410,403,397,391,384,378,371,365,359,352,346,340,333,327,321,314,308,302,295,289,283,276,270,264,257,251,244,238,232,225,219,213,206};

```
//se definen las variables que se utilizan en el programa
unsigned int16 cobid=0;
unsigned int8 lengthCAN=0;
int1 flag=0;
int8      data[6]={0x0A,0x0A,0x0A,0x0A,0x0A,0x0A},data1[6]={0x0A,0x0A,0x0A,0x0A,0x0A,0x0A},
data2[6]={0x0A,0x0A,0x0A,0x0A,0x0A,0x0A}, data3[6]={0x0A,0x0A,0x0A,0x0A,0x0A,0x0A}, am=0;
int8 balanceo1=0,balanceo2=0, pec[6]={0,0,0,0,0,0}, bal=0, ok_v=0, ocv=1, n=0, st=0;
int32 mt=0, soc=5000;
int16      v=0,          lectura=0,          ln=0,ln_prev=0,          aux=0,
voltaje[12]={0,0,0,0,0,0,0,0,0,0,0,0},t[12]={0,0,0,0,0,0,0,0,0,0,0,0},  T_min=0,  T_max=0,  T_med=0,
V_min_aux=0, V_max_aux=0;
int16 i=0, total=0, lim=0, vt=0, vis=0;
float vol=0, lo=0, aux1=0, l1=0, t_aux=0, soc_f=0;
const      int1      SELA[12]={0,1,0,1,0,1,0,1,0,1,0,1},          SELB[12]={0,0,1,1,0,0,1,1,0,0,1,1},
SELC[12]={0,0,0,0,1,1,1,1,0,0,0,0}, SELD[12]={0,0,0,0,0,0,0,0,1,1,1,1};
```

```
//variables para el calculo del CRC
```

```
int16 remainder=0, addr=0,out;
const unsigned int16 crc15Table[256] = {0x0,0xc599, 0xceab, 0xb32, 0xd8cf, 0x1d56, 0x1664, 0xd3fd,
0xf407, 0x319e, 0x3aac, //!<precomputed CRC15 Table
0xff35, 0x2cc8, 0xe951, 0xe263, 0x27fa, 0xad97, 0x680e, 0x633c, 0xa6a5, 0x7558, 0xb0c1,
0xbbf3, 0x7e6a, 0x5990, 0x9c09, 0x973b, 0x52a2, 0x815f, 0x44c6, 0x4ff4, 0x8a6d, 0x5b2e,
0x9eb7, 0x9585, 0x501c, 0x83e1, 0x4678, 0x4d4a, 0x88d3, 0xaf29, 0x6ab0, 0x6182, 0xa41b,
0x77e6, 0xb27f, 0xb94d, 0x7cd4, 0xf6b9, 0x3320, 0x3812, 0xfd8b, 0x2e76, 0xebef, 0xe0dd,
0x2544, 0x2be, 0xc727, 0xcc15, 0x98c, 0xda71, 0x1fe8, 0x14da, 0xd143, 0xf3c5, 0x365c,
0x3d6e, 0xf8f7,0x2b0a, 0xee93, 0xe5a1, 0x2038, 0x7c2, 0xc25b, 0xc969, 0xcf0, 0xdf0d,
0x1a94, 0x11a6, 0xd43f, 0x5e52, 0x9bcb, 0x90f9, 0x5560, 0x869d, 0x4304, 0x4836, 0x8daf,
0xaa55, 0x6fcc, 0x64fe, 0xa167, 0x729a, 0xb703, 0xbc31, 0x79a8, 0xa8eb, 0x6d72, 0x6640,
0xa3d9, 0x7024, 0xb5bd, 0xbe8f, 0x7b16, 0x5cec, 0x9975, 0x9247, 0x57de, 0x8423, 0x41ba,
0x4a88, 0x8f11, 0x57c, 0xc0e5, 0xcbd7, 0xe4e, 0xddb3, 0x182a, 0x1318, 0xd681, 0xf17b,
0x34e2, 0x3fd0, 0xfa49, 0x29b4, 0xec2d, 0xe71f, 0x2286, 0xa213, 0x678a, 0x6cb8, 0xa921,
0x7adc, 0xbf45, 0xb477, 0x71ee, 0x5614, 0x938d, 0x98bf, 0x5d26, 0x8edb, 0x4b42, 0x4070,
0x85e9, 0xf84, 0xca1d, 0xc12f, 0x4b6, 0xd74b, 0x12d2, 0x19e0, 0xdc79, 0xfb83, 0x3e1a, 0x3528,
0xf0b1, 0x234c, 0xe6d5, 0xede7, 0x287e, 0xf93d, 0x3ca4, 0x3796, 0xf20f, 0x21f2, 0xe46b, 0xef59,
0x2ac0, 0xd3a, 0xc8a3, 0xc391, 0x608, 0xd5f5, 0x106c, 0x1b5e, 0xdec7, 0x54aa, 0x9133, 0x9a01,
0x5f98, 0x8c65, 0x49fc, 0x42ce, 0x8757, 0xa0ad, 0x6534, 0x6e06, 0xab9f, 0x7862, 0xbdfb, 0xb6c9,
```

```
0x7350, 0x51d6, 0x944f, 0x9f7d, 0x5ae4, 0x8919, 0x4c80, 0x47b2, 0x822b, 0xa5d1, 0x6048, 0x6b7a,
0xae3, 0x7d1e, 0xb887, 0xb3b5, 0x762c, 0xfc41, 0x39d8, 0x32ea, 0xf773, 0x248e, 0xe117, 0xea25,
0x2fbc, 0x846, 0xcddf, 0xc6ed, 0x374, 0xd089, 0x1510, 0x1e22, 0dbbb, 0xaf8, 0xcf61, 0xc453,
0x1ca, 0xd237, 0x17ae, 0x1c9c, 0xd905, 0xfeff, 0x3b66, 0x3054, 0xf5cd, 0x2630, 0xe3a9, 0xe89b,
0x2d02, 0xa76f, 0x62f6, 0x69c4, 0xac5d, 0x7fa0, 0xba39, 0xb10b, 0x7492, 0x5368, 0x96f1, 0x9dc3,
0x585a, 0x8ba7, 0x4e3e, 0x450c, 0x8095};
```

//Interrupción del timer0 utilizado para el Coulomb Counting, detectar si la batería esta en reposo y el envío de datos por RS232

```
#INT_TIMER0
void intertimer0 ()
{
    set_timer0(61630);
    mt=mt+1;
    n=n+1;
    vis=vis+1;
    output_toggle(LED2);
}
```

//Interrupciones del CAN

```
#int_canrx0
void canrx0_int ( ) {
    cobid=((unsigned int16)RXB0SIDH << 3)|((RXB0SIDL & 0xE0)>>5);
```

```
    lengthCAN = (unsigned int8)RXB0DLC & 0xF;
```

```
    if(cobid == 0x080)
```

```
    {
        flag=1;
    }
```

```
    RXB0CON.RXFUL = 0;
```

```
}
```

```
#int_canrx1
```

```
void canrx1_int ( ) {
```

```
    // TODO: add CAN recieve code here
```

```
}
#int_cantx0
void cantx0_int ( ) {
    // TODO: add CAN transmit code here
}
#int_cantx1
void cantx1_int ( ) {

}
#int_cantx2
void cantx2_int ( ) {

}

//Programa principal
void main (void)
{
    //Se configura el ADC, el Timer0 y las interrupciones, se hacen operaciones de inicialización y se
    configura el CAN
    output_low(RELE);
    can_init();
    disable_interrupts(GLOBAL);

    can_set_mode(CAN_OP_CONFIG);

    BRGCON1.brp=1;
    BRGCON1.sjw=1;
    BRGCON2.prseg=2;
    BRGCON2.seg1ph=7;
    BRGCON2.sam=FALSE;
    BRGCON2.seg2phts=FALSE;
    BRGCON3.seg2ph=6;
    BRGCON3.wakfil=FALSE;
    CIOCON = 0x20;
```



```
can_set_mode(CAN_OP_NORMAL);

enable_interrupts(int_canrx0);
enable_interrupts(int_canrx1);
enable_interrupts(int_cantx0);
enable_interrupts(int_cantx1);
enable_interrupts(int_cantx2);
setup_adc_ports(ALL_ANALOG|VSS_VDD);
setup_adc(ADC_CLOCK_DIV_32);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_128);
set_timer0(61630); // overflow in 35us
enable_interrupts(INT_TIMER0);
enable_interrupts(GLOBAL);
output_high(LED1);
output_low(LED2);
for (i=0;i<1000;i++)
{
    set_adc_channel(10); //seleccionamos adc en el puerto A0
    delay_us(50); //pequeño retardo necesario
    aux=read_adc();
    lo=lo+aux/1000.0;
    delay_ms(1);
}
while(true)
{
    //orden de hacer las mediciones de voltaje
    output_low(CS);
    delay_ms(1);
    output_high(CS);
    delay_us(10);
    output_low(CS);
    spi_xfer(SP, 0x83);
    spi_xfer(SP, 0x60);
    spi_xfer(SP, 0x84);
    spi_xfer(SP, 0x78);
    delay_ms(35);
    output_high(CS);
}
```

```
//orden de leer los registros
```

```
output_low(CS);
delay_us(10);
output_high(CS);
delay_us(10);
output_low(CS);
spi_xfer(SP, 0x80);
spi_xfer(SP, 0x04);
spi_xfer(SP, 0x77);
spi_xfer(SP, 0xD6);
for(i=0;i<6;++i)
{
    data[i]=spi_xfer(SP, 0xFF);
}
output_high(CS);
```

```
output_low(CS);
delay_us(10);
output_high(CS);
delay_us(10);
output_low(CS);
spi_xfer(SP, 0x80);
spi_xfer(SP, 0x06);
spi_xfer(SP, 0xEA);
spi_xfer(SP, 0x80);
for(i=0;i<6;++i)
{
    data1[i]=spi_xfer(SP, 0xFF);
}
output_high(CS);
```

```
output_low(CS);
delay_us(10);
output_high(CS);
delay_us(10);
output_low(CS);
```

```
spi_xfer(SP, 0x80);
spi_xfer(SP, 0x08);
spi_xfer(SP, 0x2E);
spi_xfer(SP, 0x46);
for(i=0;i<6;++i)
{
    data2[i]=spi_xfer(SP, 0xFF);
}
output_high(CS);

output_low(CS);
delay_us(10);
output_high(CS);
delay_us(10);
output_low(CS);
spi_xfer(SP, 0x80);
spi_xfer(SP, 0x0A);
spi_xfer(SP, 0xB3);
spi_xfer(SP, 0x10);
for(i=0;i<6;++i)
{
    data3[i]=spi_xfer(SP, 0xFF);
}
output_high(CS);

//calculo de los voltajes de las celdas y el voltaje total
am=0;
total=0;
for(i=0;i<6;i=i+2)
{
    voltaje[am]=make16(data[i+1],data[i])/10;
    total=total+voltaje[am];
    am=am+1;
}
for(i=0;i<6;i=i+2)
{
    voltaje[am]=make16(data1[i+1],data1[i])/10;
```

```

    total=total+voltage[am];
    am=am+1;
}
for(i=0;i<6;i=i+2)
{
    voltage[am]=make16(data2[i+1],data2[i])/10;
    total=total+voltage[am];
    am=am+1;
}
for(i=0;i<6;i=i+2)
{
    voltage[am]=make16(data3[i+1],data3[i])/10;
    total=total+voltage[am];
    am=am+1;
}

//lectura de las NTCs y calculo de la temperatura máxima, mínima y media
T_max=0;
T_min=0xFFFF;
T_med=0;
for(i=0;i<12;i++)
{
    output_bit(A,SELA[i]);
    output_bit(B,SELB[i]);
    output_bit(C,SELC[i]);
    output_bit(D,SELD[i]);
    set_adc_channel(0); //seleccionamos adc en el puerto A0
    delay_us(50);      //pequeño retardo necesario
    vt=read_adc();      //leer adc
    vol=vt*5.0/1024.0+0.68;
    t_aux=(vol*10000.0)/(5.0-vol); //obtener el valor de la Req
    t_aux=(2734.115844-t_aux)/24.19940276; //obtener valor de la temperatura
    T_med=t_aux*100;
    t[i]=t_aux*100;
    T_med=T_med+(t_aux/12)*100;
    if(t[i]>T_max) T_max=t[i];
    if(t[i]<T_min) T_min=t[i];
}

```

```

}

//medida de corriente
aux1=0;
for(i=0;i<100;i++)
{
    set_adc_channel(10); //seleccionamos adc en el puerto A0
    delay_us(50);    //pequeño retardo necesario
    v=read_adc();    //leer adc
    aux1=aux1+v/100.0;
    delay_us(5);
}
lim=abs(aux1-l1);
//if(lim>3) aux1=l1;
lectura=aux1;
l1=aux1;
aux1=(aux1-l0)*5.0/1024.0;
//ln=(aux1/0.142)*1000*3.3-100;
//ln=(aux1/0.1042)*1000;
ln=32000*aux1-150;
if (ln>6500) ln=ln_prev;
ln_prev=ln;

```

//Balanceo de celdas, comprobación que los voltajes esten bien y asignación el voltaje minimo para la estimación del SoC

```

V_min_aux=0xFFFF;
V_max_aux=0;
for (i=0;i<12;i++)
{
    if(voltaje[i]>V_min && voltaje[i]<V_max && st==0)
    {
        st=0;
        ok_v=1;
    }
    else
    {

```

```

    st=1;
    ok_v=0;
}
if(voltaje[i]<V_min_aux) V_min_aux=voltaje[i];
if(voltaje[i]>V_max_aux) V_max_aux=voltaje[i];
if (i<8)
{
    if (voltaje[i]>V_bal)
    {
        balanceo1=(balanceo1 | (1<<i));
        bal=1;
    }
    else
    {
        bal=0;
    }
}
else
{
    if(voltaje[i]>V_bal)
    {
        balanceo2=(balanceo2 | (1<<(i-8)));
        bal=1;
    }
    else
    {
        bal=0;
    }
}
}

```

//orden de balanceo o de no balanceo según el resultado anterior

```

if(bal==1)
{
    pec[0]=0xF8;
    pec[1]=0;
    pec[2]=0;
}

```

```
pec[3]=0;
pec[4]=balanceo1;
pec[5]=balanceo2;
remainder = 16;
for(i = 0; i<6;i++)
{
    addr = ((remainder>>7)^pec[i])&0xff;
    remainder = (remainder<<8)^crc15Table[addr];
}
out=remainder*2;
```

```
output_low(CS);
delay_us(10);
output_high(CS);
delay_us(10);
output_low(CS);
spi_xfer(SP, 0x80);
spi_xfer(SP, 0x01);
spi_xfer(SP, 0x4D);
spi_xfer(SP, 0x7A);
```

```
spi_xfer(SP, 0xF8);
spi_xfer(SP, 0x00);
spi_xfer(SP, 0x00);
spi_xfer(SP, 0x00);
spi_xfer(SP, balanceo1);
spi_xfer(SP, balanceo2);
```

```
spi_xfer(SP, (out>>8));
spi_xfer(SP, out);
```

```
output_high(CS);
}
```

```
else
{
    output_low(CS);
```

```

delay_us(10);
output_high(CS);
delay_us(10);
output_low(CS);
spi_xfer(SP, 0x80);
spi_xfer(SP, 0x01);
spi_xfer(SP, 0x4D);
spi_xfer(SP, 0x7A);

spi_xfer(SP, 0xF8);
spi_xfer(SP, 0x00);
spi_xfer(SP, 0x00);
spi_xfer(SP, 0x00);
spi_xfer(SP, 0x00);
spi_xfer(SP, 0x00);

spi_xfer(SP, 0xBE);
spi_xfer(SP, 0xE2);

output_high(CS);
}

//activar el rele o desactivarlo dependiendo del estado de la bateria
if(ok_v==1 && ln<I_max && T_max<Tmax) output_high(RELE);
else output_low(RELE);

//estimar SoC
if (ln>500) //Coulomb Conunting si hay corriente
{
    mt=0;
    ocv=0;
    if (n>1)
    {
        n=0;
        soc_f=soc_f-(ln*0.1/3600);
        soc=soc_f;
    }
}

```



```

    }
}
if(mt>=144000 || ocv==1) //OCV si la batería esta en reposo
{
    ocv=1;
    if(T_med<=3000)
    {
        for(i=0;i<871;i++)
        {
            if(V_min_aux==V_soc_20[i])
            {
                SoC=soc_table_20[i];
                soc_f=soc;
            }
        }
    }
else
{
    if(T_med>3000 && T_med<=5000)
    {
        for(i=0;i<1544;i++)
        {
            if(V_min_aux==V_soc_40[i])
            {
                SoC=soc_table_40[i];
                soc_f=soc;
            }
        }
    }
else
{
    if(T_med>5000)
    {
        for(i=0;i<1544;i++)
        {
            if(V_min_aux==V_soc_60[i])
            {

```

```

        SoC=soc_table_60[i];
        soc_f=soc;
    }
}
}
}
}
}
}
}

```

//Cargar los buffers del CAN si se ha recibido el mensaje de sincronismo

```
if(flag==1)
```

```
{
```

```
    flag=0;
```

```
    TXB0SIDH = (unsigned int8) (0x090 >> 3);
```

```
    TXB0SIDL = (unsigned int8) (0x090 << 5);
```

```
    TXB0DLC = 8;
```

```
    TXB0D0 = make8(V_max_aux,0);
```

```
    TXB0D1 = make8(V_max_aux,1);
```

```
    TXB0D2 = make8(V_min_aux,0);
```

```
    TXB0D3 = make8(V_min_aux,1);
```

```
    TXB0D4 = make8(total,0);
```

```
    TXB0D5 = make8(total,1);
```

```
    TXB0D6 = make8(ln,0);
```

```
    TXB0D7 = make8(ln,1);
```

```
    TXB0CON.txreq = 1;
```

```
    delay_us(500);
```

```
    TXB0SIDH = (unsigned int8) (0x091 >> 3);
```

```
    TXB0SIDL = (unsigned int8) (0x091 << 5);
```

```
    TXB0DLC = 6;
```

```
    TXB0D0 = make8(SoC,0);
```

```
    TXB0D1 = make8(SoC,1);
```

```
    TXB0D2 = make8(SoC,2);
```

```
    TXB0D3 = make8(SoC,3);
```

```
    TXB0D4 = make8(T_med,0);
```

```
    TXB0D5 = make8(T_med,1);
```

```
    TXB0CON.txreq = 1;
}

//envio de datos por RS232 al ordenador
if(vis>150)
{
    vis=0;
    printf("El voltaje total es: %lu mV\r\n", total);
    printf("La temperatura media es: %lu (unidades en 0,01°C)\r\n", t_med);
    printf("La intensidad es: %lu mA\r\n", In);
    printf("El estado de carga de la batería es de: %lu (unidades en tanto por diez mil)\r\n", SoC);
    printf("-----\r\n");
}

}

}
```